Original Paper

Adaptive Behavior

Adaptive Behavior
19(1) 3–15

DOI: 10.1177/1059712310395410
adb.sagepub.com

Holistic visual encoding of ant-like routes: Navigation without waypoints

Bart Baddeley, Paul Graham, Andrew Philippides and Philip Husbands

Abstract

It is known that ants learn long visually guided routes through complex terrain. However, the mechanisms by which visual information is first learned and then used to control a route direction are not well understood. In this article, we propose a parsimonious mechanism for visually guided route following. We investigate whether a simple approach, involving scanning the environment and moving in the direction that appears most familiar, can provide a model of visually guided route learning in ants. We implement view familiarity as a means of navigation by training a classifier to determine whether a given view is part of a route and using the confidence in this classification as a proxy for familiarity. Through the coupling of movement and viewing direction, a familiar view specifies a familiar direction of viewing and thus a familiar movement to make. We show the feasibility of our approach as a model of ant-like route acquisition by learning a series of nontrivial routes through an indoor environment using a large gantry robot equipped with a panoramic camera.

Keywords

Insect navigation, route learning, view-based homing, image classification, autonomous robotics

1 Introduction

Individual ant foragers show remarkable navigational performance, rapidly learning long idiosyncratic routes through cluttered environments (Cheng, Narendra, Sommer, & Wehner, 2009). While the initial stages of route acquisition are underpinned by the ants' path integration system, as ants become more familiar with a given route, so the use of visually mediated navigational strategies comes to the fore (Collett, Dillmann, Giger, & Wehner, 1992; Durier, Graham, & Collett, 2003; Rosengren & Fortelius, 1986; Wehner, 1996; Wehner & Radber, 1979). These and similar studies of visual navigation have revealed how insects combine simple strategies to produce robust behavior. This has established insect navigation as a model system for investigating the sensory, cognitive, and behavioral strategies that enable animals to perform complex behaviors in the real world.

One elegant use of visual landmark information is view-based homing. Behavioral experiments with ants (Durier et al., 2003; Wehner & Radber, 1979) and bees (Cartwright & Collett, 1983) have shown that individuals store two-dimensional retinotopic views of the world as seen from their goal location. Subsequent search for that goal location can be driven by a comparison of their current view of the world and the view stored at the goal (Franz, Schölkopf, Mallot, & Bülthoff, 1998). Computational studies have shown that this tactic is successful within a catchment area centered on the goal, the size of which depends on the depth structure of the world (Stürzl & Zeil, 2007; Zeil, Hofmann, & Chahl, 2003). However, as this can be an efficient and economical mechanism, it is not a great leap to imagine that navigation over larger scales, that is, along routes, could be achieved by internalizing a series of stored views linked together as a sequence. Route behavior in this framework would entail homing from one stored view to another.

Centre for Computational Neuroscience and Robotics, School of Informatics, University of Sussex, UK.

Corresponding author:
B. Baddeley, Centre for Computational Neuroscience and Robotics, School of Informatics, University of Sussex, Brighton, UK
Email: bartbaddeley@googlemail.com

Recent studies of ants suggest, however, that route-guidance could be performed using simpler procedural rules whereby the heading governing a path segment becomes associated with an appropriate visually identified location (Collett, Collett, Bisch, & Wehner, 1998; Graham & Cheng, 2009). Moreover, attempts to model route navigation using linked view-based homing have shown it to be a nontrivial problem which requires the agent to both robustly determine at which point a waypoint should be set during route construction and decide when a waypoint has been reached during navigation (Franz, Schölkopf, Georg, Mallot, & Bülthoff, 1998; Smith, Philippides, Graham, Baddeley, & Husbands, 2007; Vardy, 2006). Essentially, for robust route navigation within the framework of a sequence of snapshots, an agent needs place recognition to determine where along the route it is (Smith, Philippides, Graham, & Husbands, 2008).

Our study of visually guided routes therefore takes a different tack. Instead of defining routes in terms of discrete waypoints, we define a route-learning process in which the route is learned more holistically. In this framework, the agent employs a classifier that predicts whether a given view is on or off the learned route. A behavioral routine then facilitates route following by scanning the world and moving in the direction that is deemed most likely to be part of the route.

We feel that this approach has two main benefits. First, a classifier is a parsimonious way to encode a series of views. We do not attempt to learn every view along the route, but instead use them to learn the classifier. By using a classifier to determine whether a given view comes from part of the learned route or not, our approach provides a compact way of storing the visual information required to follow routes. As a corollary, the agent does not need to decide when or which views to learn. Second, the classifier we choose to use also outputs a confidence in the classification it has made. The classifier can therefore be applied to any view to determine the degree of confidence that it is part of the route. One can think of the confidence as being a proxy for how familiar that view is. Ants can only translate in one direction relative to their viewing direction, namely forward. This tight coupling of sensation and action allows us to re-frame the problem of navigation in terms of a search for the views that are associated with a route. By visually scanning the environment and moving in the direction that is most similar to the views encountered during learning an ant or robot should be able to reliably retrace a given route.

Note that this process associates the current view *not* with a particular place but instead with a particular action, that is, "what should I do?" not "where am I?"

Both desert ants and wood ants perform scanning behaviors that would support this approach. When released in an unexpected but familiar place the desert ant *Melophorus bagoti* scans the environment by turning rapidly on the spot [P. Graham, personal observation]. More than one scan may be performed with short straight runs of a few centimeters separating them before the ant finally sets off in a seemingly purposeful manner. Wood ants exhibit a second form of scanning behavior. Instead of walking in a straight line, they instead tend to weave a somewhat sinuous path (Graham & Collett, 2002). This has the effect of producing scans of the world centered on the overall direction of movement.

We implement this approach on a real robot navigating a series of nontrivial routes through visually cluttered environments. Our results indicate that it is possible to implement a holistic, view-based homing strategy for acquiring ant-like routes. Crucially, there is no need to break the route up into a series of discrete waypoints, instead learning occurs continuously. This avoids the problem of determining what should trigger the learning of a new view or waypoint and leads to a simpler mechanism. Moreover, we suggest this approach provides a powerful platform for investigating efficient encoding of route guidance information and how this depends on the visual ecology within which an agent navigates.

2 Methods

2.1 Overview

In order to test our hypothesis we need to sample the world from an ant's view point. To do this we use a large volume Cartesian XYZ gantry robot to sample panoramic images along a prespecified ground-level trajectory through a cluttered environment. Images are collected facing forwards and also at angles of $\pm 45^\circ$ relative to the route heading. The circular panoramic images are unwrapped in software to produce rectangular images with a resolution of 4°/pixel. A pool of 5,000 simple block-like feature detectors are randomly initialized and used to form the basis of our image representation. Each feature detector forms a simple classifier by determining a threshold above which it classifies the image as one class and below which it classifies it as the opposite. Learning then involves selecting features from the pool and determining how to set their thresholds and weight their different predictions in order to form a final robust classification. The learned classifier is used to recapitulate the learned route using the gantry robot in a closed-loop mode. Each of these stages will now be described in more detail.

2.2 Data collection

All experiments reported here were performed on a gantry robot—a large volume XYZ Cartesian robot (Figure 1a). The gantry axis configuration provides an operating volume of 3,000 mm $\times$ 2,000 mm $\times$ 2,000 mm. The sensor end of the Z-axis can be placed anywhere within this volume with sub-millimeter accuracy. For the experiments presented here a catadioptric camera system (VCAM 360) is mounted on the z-axis to produce panoramic images. A panoramic mirror projects a 360° image of the environment onto a downward facing CCD video camera. The image is transformed from a circular reflection (Figure 1b) into a panoramic image that is used for subsequent processing. This process is performed in software using cubic interpolation and results in a [90×29] rectangular panoramic image with a resolution of 4°/pixel (Figure 1c). This value represents our best guess for the ants' visual acuity.

The gantry workspace was populated with a variety of objects consisting of foam blocks, piles of fabric, paper rolls and a random selection of toys. Objects were placed in such a way that it was possible to move the sensor head along a route through this visual clutter. Routes could be made more or less challenging by varying the degree of clutter and the straightness of the routes. In order to go beyond what is possible with a snapshot type model using a single snapshot, the beginning and end points of some routes were chosen so that it would not be possible to perform the route using this approach. This is achieved by making sure that the end point of the route could not be viewed from the starting position.

In order to train a classifier it is necessary to generate positive and negative training examples of the input to be classified. In our case this means collecting views that are part of the route and views that are not part of the route. The positive examples are simply the forward-facing views experienced along the route. The negative views consisted of views from the route taken facing to the left and right of the direction of movement at an angle of ±45° relative to the route heading (Figure 1d). A small amount of normally distributed noise (standard deviation = 6°) was added to each of the sampling directions. This approach is inspired by the observation that ants tend not to move in a straight line on a route but instead proceed in a sinuous

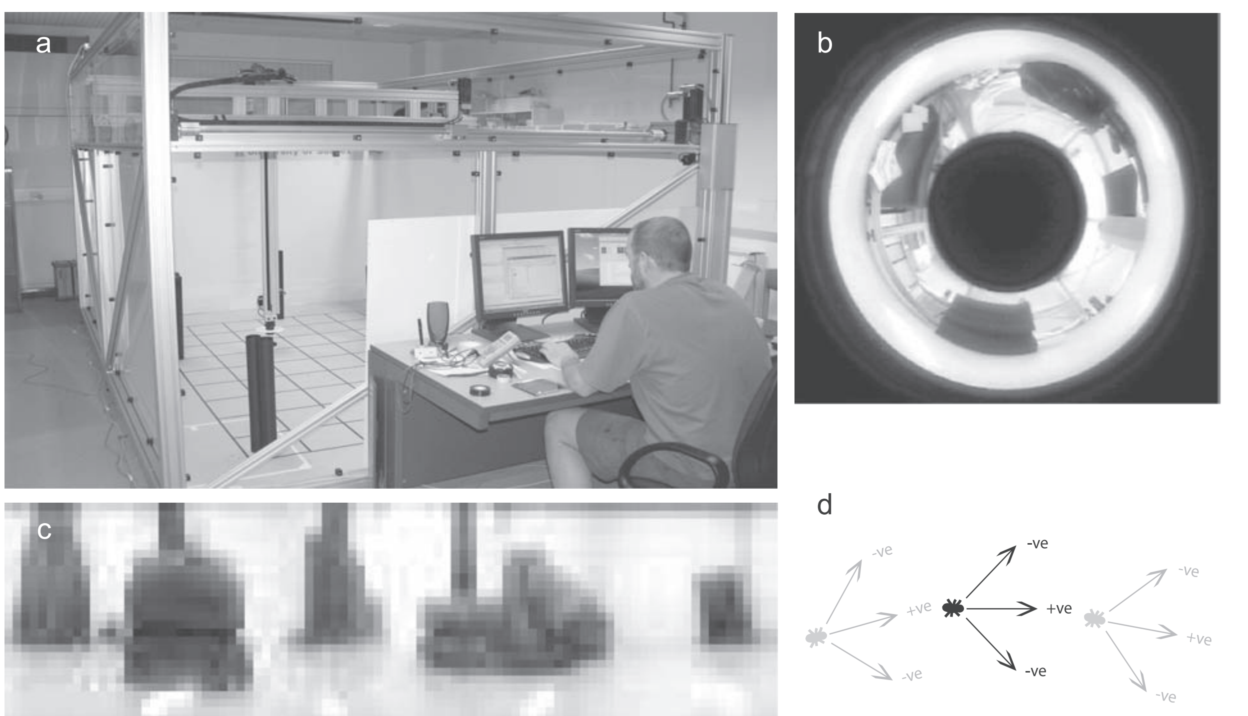

Figure 1. Data Collection. (a) The gantry robot used in all experiments. (b) Example raw panoramic image prior to unwrapping in software. (c) Example unwrapped [90×29] rectangular image representing a resolution of 4°/pixel. (d) Images are collected in three directions for each point along the training route. A forward facing image is collected as an example of an on route view and two off route views are also collected at ±45° relative to the route heading.

manner that results in some views that do not relate to the overall direction of travel and some that do.

2.3 Image representation

Classifying images is a difficult task because of the high dimensionality of the input if one adopts a pixel-by-pixel representation. In order to make learning tractable one needs to project this high-dimensional space into a lower dimensional space that retains enough of the necessary structure to allow successful classification of the input. As a first step in reducing the dimensionality of the input we downsampled the images to a resolution of 4°/pixel. To control the dimensionality of the input further we project the image into an N-dimensional feature space using a pool of simple Haar-like feature detectors (Papageorgiou, Oren, & Poggio, 1998).[1]

Feature detectors were selected from a randomly initialized pool of 5,000. Initialization involved randomly selecting any two image coordinates that define the opposite corners of an image patch representing the spatial extent of the feature detector. Having defined the size and position the feature detector is then randomly assigned to one of six possible classes (see Figure 2). The value of a *one-rectangle feature* is simply the mean intensity value of the patch. The value of a *two-rectangle feature* is the difference between the mean intensity of two rectangular regions. The regions have the same size and shape and are located next to each other either horizontally or vertically. The value of a *three-rectangle feature* is given by the mean intensities of two outer rectangles subtracted from the mean intensity of a central rectangle, again oriented either horizontally or vertically. Lastly, a *four-rectangle feature* computes the difference between diagonal pairs of rectangles. Example features are shown in Figure 2. The features act like edge detectors or crude approximations to Gabor filters (Gabor, 1946) and are maximally activated at high contrast boundaries in the image. The output of these feature detectors forms the basis of our image representation (Figure 3).

2.4 Boosting

There are many different approaches to learning a classifier that we could have employed, although one prerequisite for our approach is that the classifier should provide a measure of the confidence in its predictions. Following Viola and Jones (2001) we chose to construct a boosted classifier using Haar-like features as this approach provides us with a confidence in our predictions, as well as providing a way to control the complexity of the learned classifier by prespecifying the number of Haar-like features that the classifier employs.

We now describe how to use boosting to build a classifier that can be used to recognize views associated with a learned route. Boosting is a supervised learning technique for constructing a *strong classifier* from a set

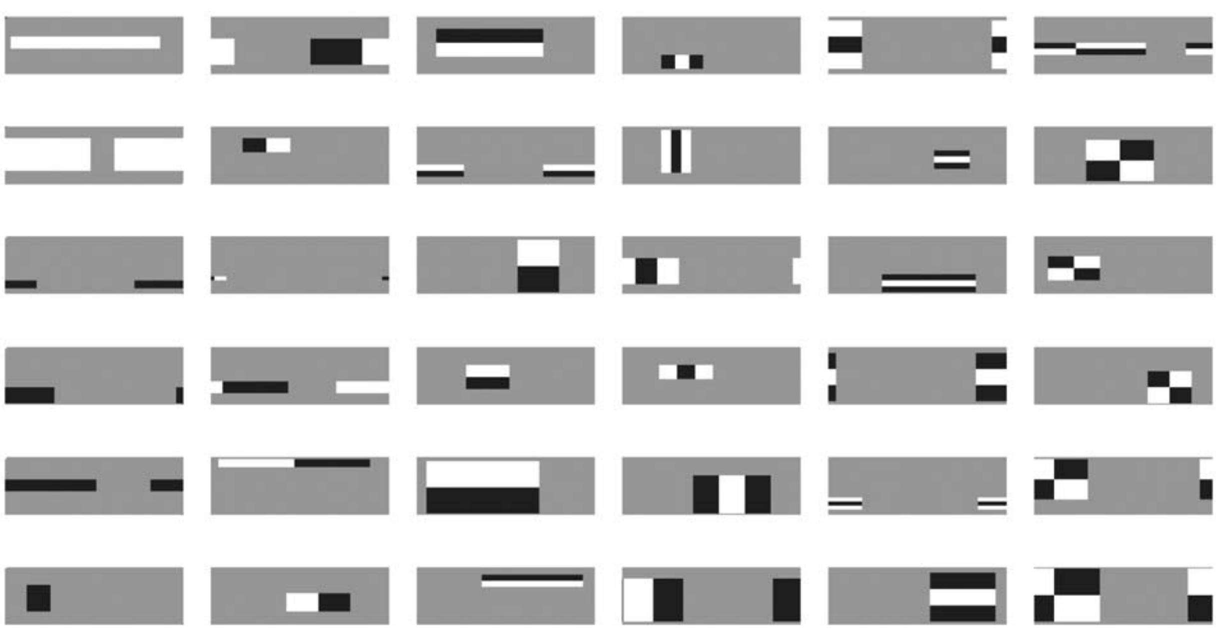

Figure 2. Examples of the six different classes of Haar-like feature detector. Each column shows examples of one of the six different classes of feature detector. Note each class of feature can vary in size, shape, and position. Note also that feature detectors wrap-around if they extend beyond the left or right edge of the image.

of *weak classifiers* given a training set of labeled positive and negative examples. A weak classifier is one that performs only slightly better than chance. Conversely, a strong classifier is one that performs arbitrarily well. A strong classifier is constructed from a linear weighted combination of the outputs of weak classifiers.

There exist many variants of boosting algorithms. Adaboost (Freund & Schapire, 1995), the approach we use in this article, is one of the most commonly used. The basic algorithm works as follows. At each iteration, the training data are resampled or reweighted according to a distribution of weights that indicate the current importance of each example in the dataset. A weak classifier is then learned using this resampled/reweighted dataset and is added to the strong classifier. The relative contribution of each of the weak classifiers to the final strong classifier is determined by performance on the sampled data. Finally, the weights of incorrectly classified examples are increased and correctly classified examples decreased, thereby encouraging the next weak classifier to focus more on the examples that were incorrectly classified at the last iteration. Weak classifiers are added until the overall classification performance exceeds some threshold or the maximum number of weak learners is reached.

The pseudocode for Adaboost is as follows:

Set $T =$ maximum number of weak classifiers
Given: $(x_1, y_1), \ldots, (x_m, y_m)$
where $x_i \in X$, are the outputs of the feature detectors and $y_i \in Y = \{-1, +1\}$ are the class labels of the training set

Initialize $W_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

Find the classifier $h_t : X \to \{-1, +1\}$ that minimizes the error with respect to the distribution W_t
Calculate ε_t the weighted error rate of classifier h_t with respect to the reweighted data at time t.
If $\varepsilon_t = 0$ then break
Choose $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$
Update $W_{t+1}(i) = \frac{W_t(i) \exp(-\alpha_t \cdot y_i \cdot h_t(x_i))}{Z_t}$ where Z_t is a normalization factor that ensures that W represents a probability distribution over the training data

Output the final classifier $H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Following Viola and Jones (2001) we implement Adaboost using single Haar-like features as the basis of our weak classifiers. A weak classifier $h_j(x)$ thus consists of a Haar feature f_j, a threshold θ_j, and a parity p_j that determines whether the output of the feature detector f_j should be greater than or less than the threshold θ_j in order that the input be classified as positive. The process by which a weak classifier is constructed is illustrated in Figure 4.

$$h_j(x) = 1 \quad \text{if} \quad p_j f_j(x) < p_j \theta_j$$
$$h_j(x) = 0 \quad \text{if} \quad p_j f_j(x) > p_j \theta_j$$

By providing a pool of feature detectors, each defining a weak learner, Adaboost is able to perform feature selection. At each iteration a single feature detector is

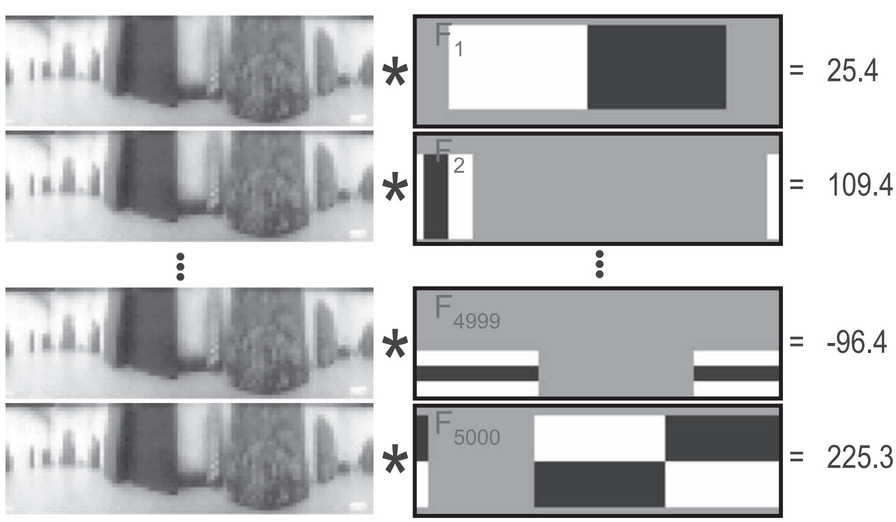

Figure 3. Image representation. Each image is represented by the output of the 5000 randomly initialized feature detectors applied to the image. Note these 5000 features represent the pool that is selected from. The final classifier utilizes only a small subset, between 5 and 200, of this total. The idea of using this alternative to a pixel-wise representation was first explored by Papageorgiou et al. (1998), who used the outputs of Haar-like features to categorize images.

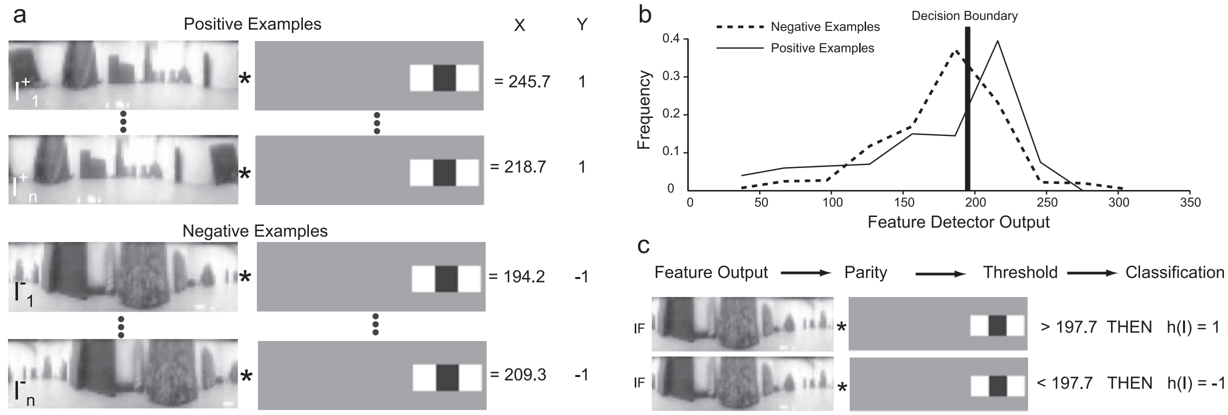

Figure 4. Constructing a weak classifier using a single Haar-like feature. (a) Each of a set of training images are convolved with the feature detector resulting in two sets of real numbers, one representing positive instances and one representing negative instances. (b) Frequency histogram showing the distribution of feature detector outputs for the two classes. A threshold is determined that best separates the two classes. (c) The final weak classifier is defined by a feature detector, a threshold, and a parity. For the example shown, the output of the weak classifier, $h(I)$, for a given image, I, is equal to 1 if the feature detector output exceeds a threshold of 197.7 otherwise it is equal to -1.

chosen that best aids in classification. This allows Adaboost to pick out and use only those features that are most useful for the current classification problem.

Key to our use of a boosted classifier is the fact that it is possible to obtain a confidence value associated with any given classification made using the trained classifier. This confidence value is related to the margin and is given by:

$$conf(x) = \left\| \sum_{t=1}^{T} \alpha_t h_t(x) \right\|$$

Which is simply the degree to which the sum of the combined weak classifiers differ from zero, prior to the sign being taken.

By applying the classifier to views in different directions we can attempt to determine which of the views are from the learned route (Figure 5). By weighting each of the viewing directions that produce positive classifications by their associated confidence values we can determine a direction to move that is most likely to keep us on the learned route.

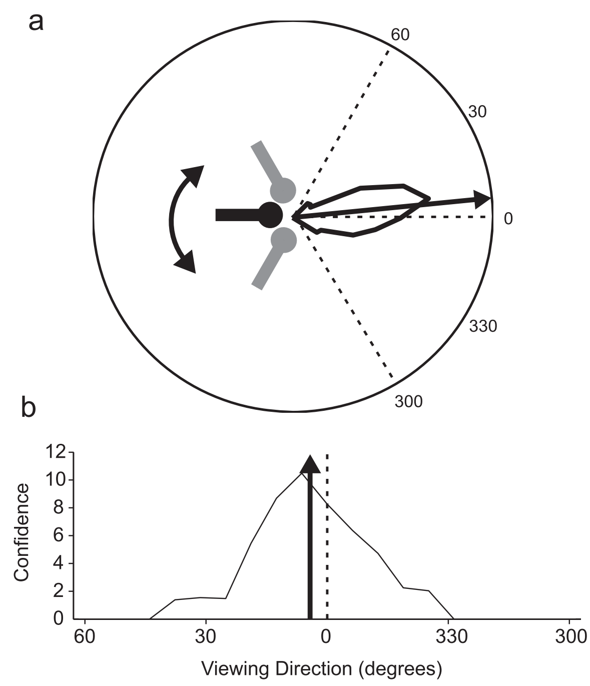

Figure 5. Determining the direction of travel using a scanning routine. (a) Images are sampled in a range of directions from -60° to $+60^\circ$ in steps of 5° relative to the current heading. Each image is classified and if the classification is positive then the confidence in the classification is used to weight the viewing direction associated with that view. The figure shows a polar plot of the confidence values for all positively classified viewing directions. The weighted average of the positively classified viewing directions is represented by the long arrow. (b) The same information represented using a standard plot. The dashed line indicates the current heading and the solid line represents the weighted average of the positively classified viewing directions.

2.5 Route following using a boosted classifier

To test the system's performance, Haar-like features are extracted from a set of training images collected every 5 cm and used to train a boosted classifier. During testing the camera is positioned at the start of the route facing in the correct direction. From this position images are sampled in a range of directions from -60° to $+60^\circ$ in steps of 5° relative to the current heading. Features are extracted from all of these images and used as input to the classifier. All of the viewing directions that produce a positive classification contribute to a weighted average with the weighting controlled by the confidence interval of the individual classifications, $\phi_{final} = \sum conf(i) \times h(i) \times \phi(i)$, where ϕ_{final} is the ultimate direction of travel, $conf(i)$ and $h(i)$ are the confidence and classification that are output by the classifier relating to the ith view of the scan, and $\phi(i)$ is the viewing direction for this view. This weighted average is then used to determined the direction of travel and a 5 cm step is made in this direction. The process is then iterated until success or failure.

3 Results

In order to test our approach we performed a series of three experiments. In each instance a classifier was trained using a set of images gathered during a single traversal of a prespecified training route. Performance was then assessed by starting the robot in a series of different positions close to the original starting point and running the algorithm for a number of steps just larger than the number of steps used in the training phase. This allowed for the possibility of runs that reached the goal while taking a slightly longer route. In fact, most of the time the learned routes were more direct than the training routes as can be seen in Figures 6, 7, and 8.

As described in the methods, positive and negative instances for training the classifier were generated from views along the route in the direction of movement for positive instances and the same views rotated by $-45°$ and $+45°$ for the negative instances. Haar-like features were extracted from these views and used to train a boosted classifier. For each of the routes we trained classifiers with differing numbers of weak classifiers, each instantiated using a single Haar-like feature selected from a pool of 5,000 randomly initialized features.

In the first set of experiments we investigated the problem of learning a route from one corner of the gantry's workspace to the opposite corner, a distance of approximately 3 m. For these experiments all objects were placed along the inside walls of the gantry and outside of the working limits of the moving sensor head. Example views along a straight path are shown in Figure 6. Despite the route's simplicity there is still a large variation in the views associated with it. In order to vary the task difficulty, training routes were generated with increasingly tortuous paths, starting with a completely straight path and ending up with a path that included loops and turn backs. Because the positive and negative views were always collected in the same way, these later paths inevitably included positive and negative views that were very similar resulting in a classification problem that could not be solved perfectly. Collecting training images every 5 cm resulted in 60 positive and 120 negative views that were used to train the classifier. Figure 6 shows the results for three different routes and boosted classifiers with 5, 10, and 20 weak classifiers or Haar-like feature detectors. The top part of the figure shows examples of views from the route together with the output of the classifier for scans from $-60°$ to $+60°$ superimposed on top. The classifier output represents the confidence in the classification if the classification is positive, otherwise it is zero. The confidence values have been scaled for the purposes of illustration and are in fact higher for the classifiers with greater numbers of feature detectors. As the number of feature detectors is increased the transition from viewing directions producing negative classifications to those producing positive ones becomes smoother as does the overall profile of the confidence estimates. The lower part of Figure 6 shows the performance on training routes of increasing complexity. In all cases performance is generally good with a path from corner to corner of the workspace being successfully learned. As the training routes become more tortuous performance becomes less consistent with higher numbers of weak classifiers resulting in slightly more direct and stable paths.

In a second set of experiments a large pile of foam blocks covered by a gray sheet was placed in the center of the workspace. This required that we define this central region as out of bounds to the robot to prevent damage to the panoramic imaging device (dark region in Figure 7b and c). A circuit of the workspace was then defined which resulted in a training route that passed around the obstacle. Collecting training images every 5 cm resulted in 210 positive and 420 negative views that were used to train the classifier. In this instance the number of classifiers required to completely learn the route was far higher. Although reasonable performance was still achieved using just five features, the paths tended to get stuck in a loop circling the obstacle (Figure 7b). We found it necessary to increase the number of features used to 50 in order to get reliable performance (Figure 7c). Figure 7d shows confidence maps indicating how the confidence in positive classifications varies with respect to time step and viewing direction. In the left-hand figure, representing the performance of the five-feature classifier, the robot fails to straighten its path and complete the route and instead gets drawn back into another loop at the point indicated by a star. In the right-hand figure we see how the 50-feature classifier performs. The star indicates the point where the five-feature classifier fails, we see that although confidence is reduced at this point the 50-feature classifier is still able to correctly straighten the path and complete the route.

In our final experiment we used our approach to learn an S-shaped path through a cluttered environment. Figure 8 shows the results for this environment.

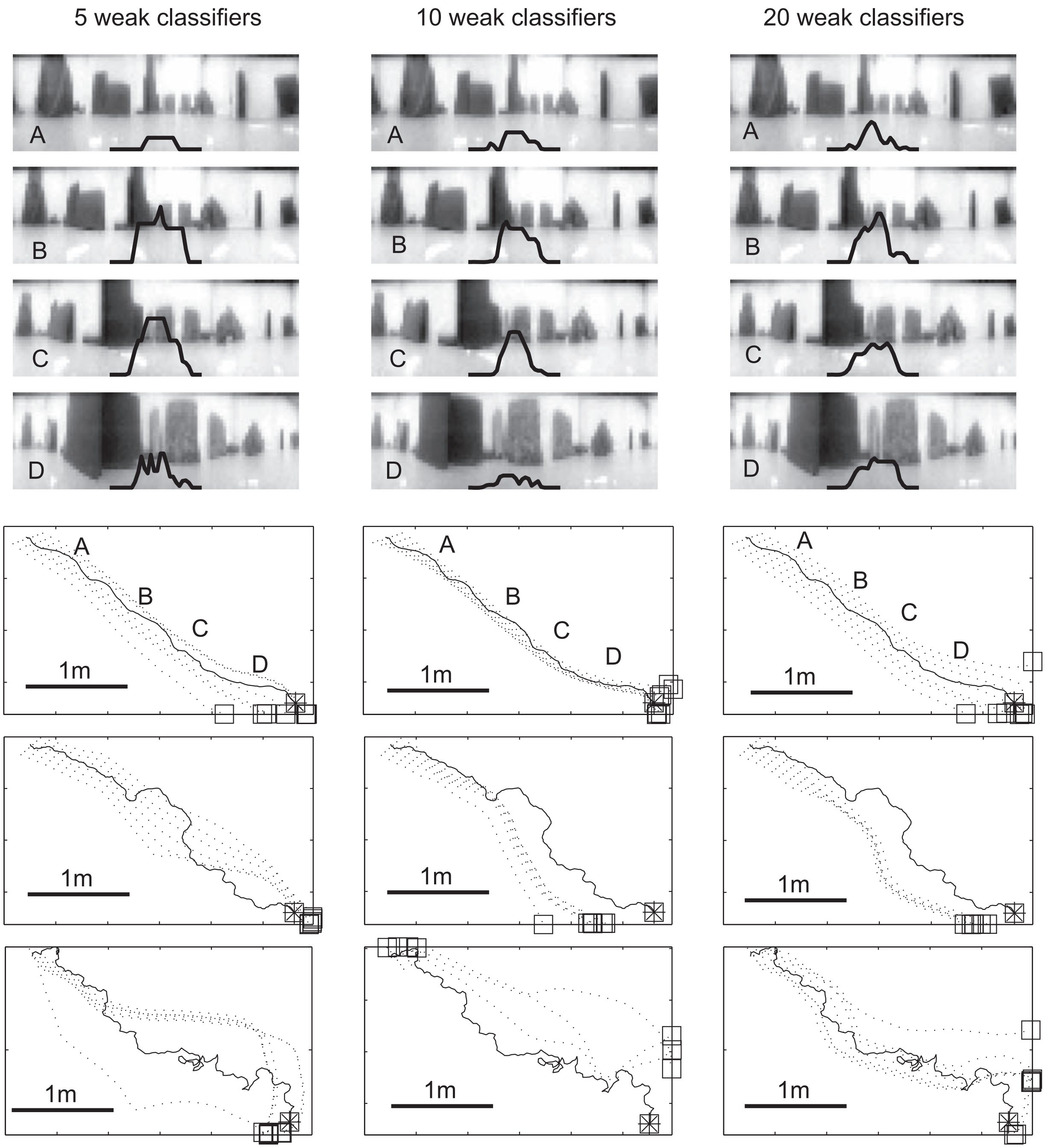

Figure 6. Example routes across the empty workspace. Top: Example views and classifier output (black line) for scans from -60° to $+60^\circ$ centered on each of the views shown. The letters A, B, C, and D relate to positions indicated in the plots below. The classifier output represents the confidence in the classification if the classification is positive, otherwise it is zero. Confidence values have been scaled for the purposes of illustration. Bottom: Three different routes of differing straightness, learned using boosted classifiers with differing numbers of Haar-like features. The letters A, B, C, and D refer to the approximate positions where the views above were captured. The solid lines show the training route and the dotted lines indicate test runs from different starting points. Squares indicate the endpoints of the test runs.

Collecting training images every 5 cm resulted in 180 positive and 360 negative views that were used to train the classifier. As in the previous example it was not possible to learn the route using very small numbers of features and again good performance was only achieved with a boosted classifier with 50 features. For comparison, the original work by Viola and Jones (2001) on which our classification approach is based employed 200 features to detect and classify faces in 384×288 pixel images.

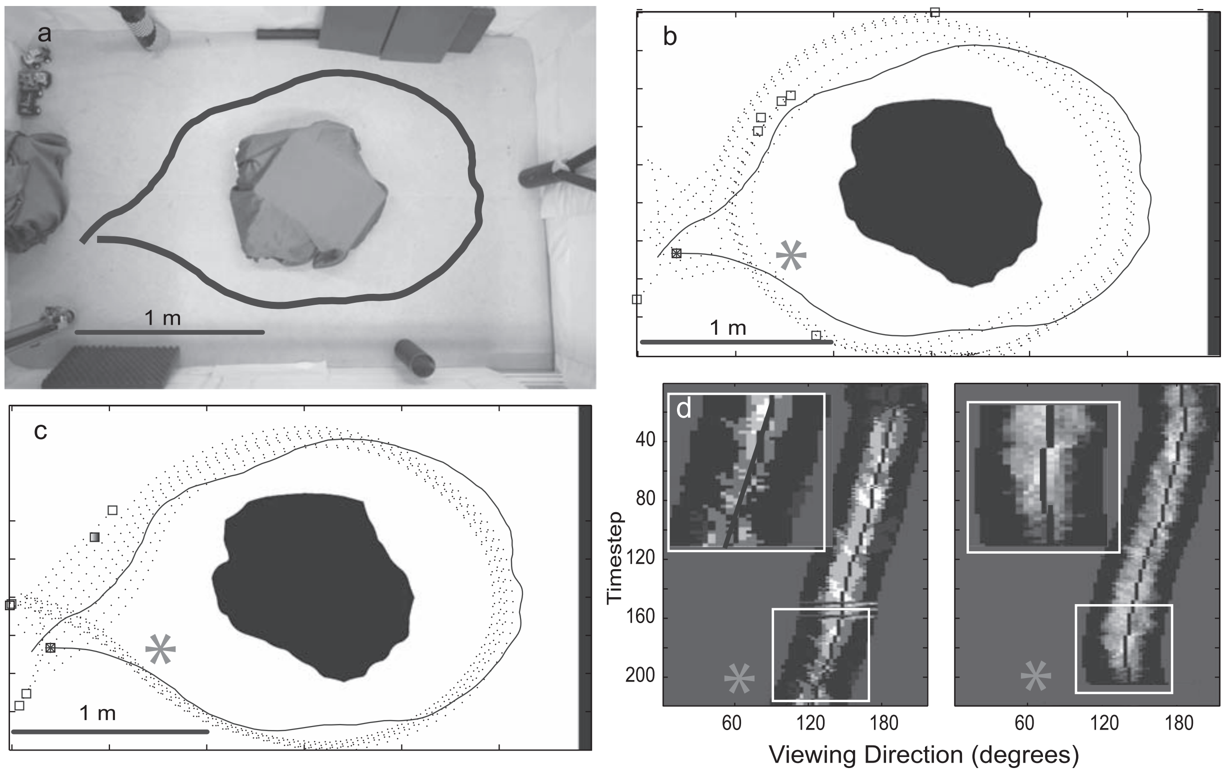

Figure 7. Learning a circuit of the workspace. (a) The gantry workspace as viewed from above. The dark line represents the training route. (b) Performance of the algorithm using a boosted classifier with five features. At the point indicated by the star the robot fails to complete the route and instead is drawn back into another loop. (c) Performance of the algorithm using a boosted classifier with 50 features. In this instance the robot is able to correctly avoid being drawn back into the loop at the point indicated by the star. (d) Confidence maps indicating how the confidence in positive classifications varies with respect to time step and viewing direction. Insets show a zoomed-in final section where the two plots differ and the performance of the two trials varies. Each horizontal slice represents a single scan that is limited to a range of -60° to $+60^\circ$ relative to the current viewing direction. Time increases from top to bottom. In this instance there is a clear trend indicating clockwise rotation, as would be expected for a loop. Directions outside of the scanning range are uniform gray and the weighted average at each time step is indicated by the dark line in the main figure and in the inset. In the left-hand figure, representing the performance of the five-feature classifier, the robot fails to straighten its path and complete the route and instead gets drawn back into another loop at the point indicated by a star. In the right-hand figure we see how the 50 feature classifier performs. The star indicates the point where the five-feature classifier fails, we see that although confidence is reduced at this point the 50-feature classifier is still able to correctly straighten the path and complete the route.

4 Discussion

4.1 Summary

We have shown that it is possible to learn a nontrivial route through an environment using a simple view-classification strategy based on positive and negative views collected during a single episode of learning. By considering the tight coupling of sensation and action that is present in ants and some robots, we were able to re-frame the problem of route navigation in terms of a search for familiar views using a classifier that provides a compact way of storing the information required to recognize views and, crucially, a measure of the expected uncertainty of the classification.

The idea that routes can be learned using a set of Stimulus–Response (S–R) relations is not new (Gaussier & Zrehen, 1995; Giovannangeli, Gaussier, & Désilles, 2006). Equally, it has been observed by various authors that it is possible to orient rotationally by comparing views in different directions to a reference view, effectively resulting in a visual compass (Graham, Philippides, & Baddeley, 2010; Philippides, Baddeley, Cheng, & Graham, in press; Zeil et al., 2003). However, combining aspects of these two approaches, as we have done, constitutes a novel approach. Firstly, by parameterizing the S–R relationship using a boosted classifier, we not only provide a compact representation of the problem, we also obtain a less brittle solution by being less reliant on determining an exact match between the learned stimulus and the current view.

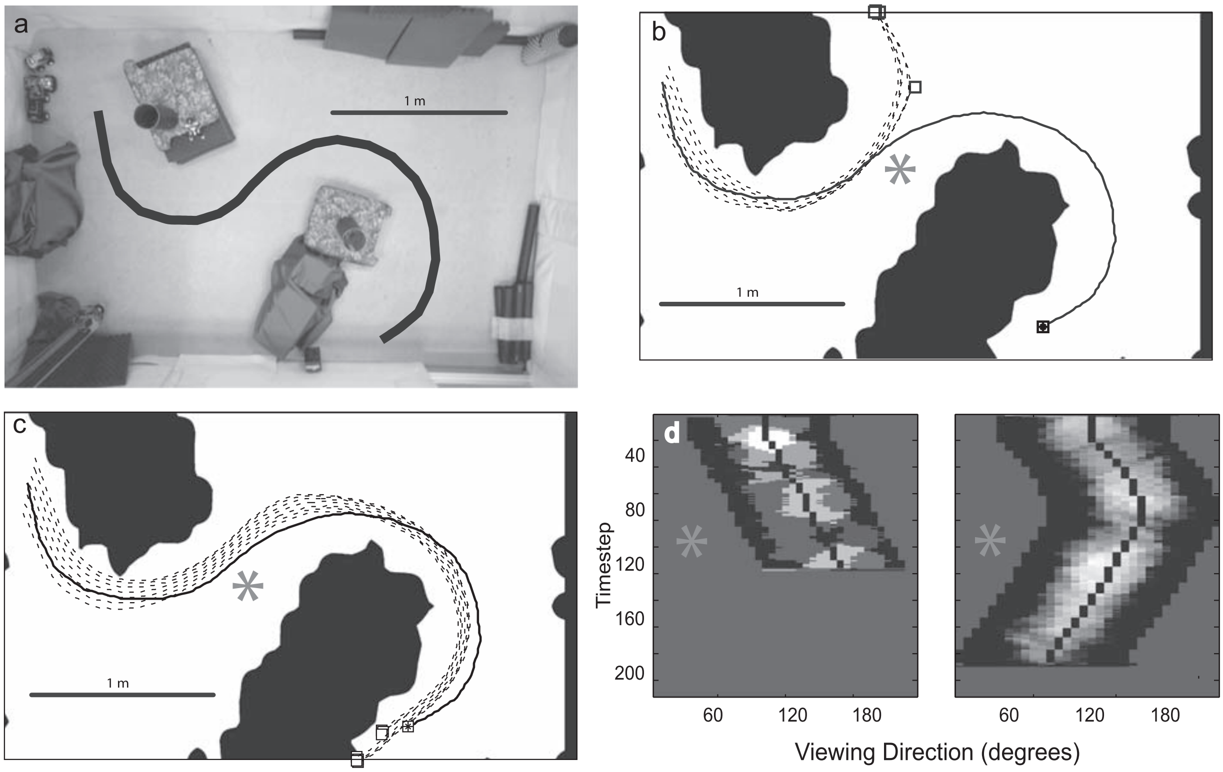

Figure 8. Learning a route through clutter. (a) The gantry workspace as viewed from above. The dark line represents the training route. (b) Performance of the algorithm using a boosted classifier with five features. The robot completes the first section of the route but fails to reverse turning direction at the appropriate time. (c) Performance of the algorithm using a boosted classifier with 50 features. The robot completes the route with a high degree of consistency. (d) Confidence maps indicating how the confidence in positive classifications varies with respect to time step and viewing direction. Each horizontal slice represents a single scan that is limited to a range of -60° to $+60^{\circ}$ relative to the current viewing direction. Time increases from top to bottom. In this instance there is a clear trend indicating counter-clockwise rotation for the initial section followed by clockwise rotation in the right-hand plot. Directions outside of the scanning range are uniform gray and the weighted average at each time step is indicated by the dark line. In the left hand figure, representing the performance of the five-feature classifier, the robot fails to reverse direction at the appropriate time indicated by a star. In the right hand figure we see how the 50-feature classifier performs. The star indicates the point where the five-feature classifier fails, we see that although confidence is reduced at this point the 50-feature classifier is still able to correctly reverse direction and complete the route.

Secondly, by using the classifier to determine view familiarity we are performing recognition rather than recall, which is a fundamentally easier problem. In using familiarity rather than similarity to a particular reference view we can go beyond a simple visual compass and instead end up with a method for learning entire routes.

By embodying the view classifier on a physical platform and constraining the required spatial behavior to routes, we were able to explore other areas for parsimony. Because each scan is centered about the current heading, the same position in space can elicit different responses when approached from different directions. As long as the agent has some context provided by the likely starting direction of travel and scans the environment over a limited range of directions relative to its current heading, it can recapitulate a learned route through a visually cluttered world and produce sensible headings from points off the original learned route. This provides an interesting example of where a simple interaction between a behavioral strategy and learned information provides robust behavior. Without such an interaction the agent would require a much more comprehensive survey of the environment. Interestingly, this type of interaction has been observed in ants where directional information from path integration has been shown to increase the precision of visual landmark use (Fukushi & Wehner, 2004).

One potential issue regarding our proposed approach is the problem of visual aliasing. If the world looks similar at two different locations and requires different actions to be performed then we can expect the process to fail at one of the locations. While this is a problem for a general route-learning algorithm we believe it to be less of a problem for a model of route learning in ants. That is, we would also expect the ants to have problems learning such a route. In practice it is likely that visual aliasing is rarely a problem in natural environments and in situations where visual information is ambiguous we would expect that other sensory modalities would allow the disambiguation of the two locations.

4.2 Relating the model to ant behavior

Our ultimate goal with this project is to understand likely and viable mechanisms used by insects for navigation. Therefore it is useful to summarize our framework with respect to some of the desirable properties of insect route behavior: (a) Route knowledge should be procedural, that is, an agent should be able to produce the correct behavior for a given place independently of the prior sequence of visited places. By constraining vision and motion we produced a simple procedural mechanism for visually setting heading that is independent of the sequence of prior visited places given the current heading. (b) Route knowledge should consist of a broad corridor of familiar places rather than a fragile narrow ridge and agents need to produce sensible behavior when they are outside the route corridor. It is an open question as to whether ants are drawn back into their habitual routes when they are far outside of their route corridor (Collett, Graham, & Harris, 2007) and there is anecdotal evidence that they may not recognize a familiar route if approached from an unfamiliar direction [Michael Mangan, personal communication]. However, in most instances they will rejoin a route if they happen across it while searching (Kohler & Wehner, 2005; Wehner, Boyer, Loertscher, Sommer, & Menzi, 2006). We observe similar behavior using our approach and the estimates of heading produced from close off-route locations are sensible, normally being parallel to the route or drawing the robot slightly in towards the route. As one moves further from the route the uncertainty in recalled headings (signified by lower confidence in the output of the classifier) increases, which would be a useful signal to commence a systematic search for the route; a behavior seen in ants when they are lost (Kohler & Wehner, 2005).

4.3 Prospects

We have presented a proof of concept of a simple view-classification system that can take advantage of a tight sensorimotor coupling to produce a route-navigation system. Our goal was to demonstrate that routes can be represented holistically thus allowing route recapitulation to be described as a recognition problem. Therefore, to some extent it is the mature route encoding which provides the proof of concept. In future work we will address issues relating to how learning might be achieved without the need for off-line training of the classifier.

Although there is inherent value in such a parsimonious model, it is of potentially greater interest that it provides a framework within which we can investigate the three-way sensory–motor–environment interaction that shapes behavior. For instance, we suggest that this model can be used to investigate which visual primitives might best be used for visual route guidance in natural environments. Similarly, we can explore the interactions between movement strategies, learning, and route performance. Because we are trying to gain insight into how ants might use visual information to guide their routes, rather than attempting an engineering solution to the problem of visual navigation, successful route following is only a minimum requirement. Of greater interest are insights into how limitations and constraints can shape the route-learning process.

Funding

This work was supported by the Biotechnology and Biological Sciences Research Council [grant number BBF0100521].

Note

1. Simple block-like features with similarities to the Haar wavelet (Haar, 1910).

References

Cartwright, B. A., & Collett, T. S. (1983). Landmark learning in bees. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, *151*(4), 521–543.

Cheng, K., Narendra, A., Sommer, S., & Wehner, R. (2009). Traveling in clutter: Navigation in the central Australian desert ant *Melophorus bagoti*. *Behavioural Processes*, *80*(3), 261–268.

Collett, M., Collett, T. S., Bisch, S., & Wehner, R. (1998). Local and global vectors in desert ant navigation. *Nature*, *394*, 269–272.

Collett, T. S., Dillmann, E., Giger, A., & Wehner, R. (1992). Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, *170*(4), 435–442.

Collett, T. S., Graham, P., & Harris, R. A. (2007). Novel landmark-guided routes in ants. *Journal of Experimental Biology*, *210*(12), 2025–2032.

Durier, V., Graham, P., & Collett, T. S. (2003). Snapshot memories and landmark guidance in wood ants. *Current Biology*, *13*(18), 1614–1618.

Franz, M., Schölkopf, B., Georg, P., Mallot, H., & Bülthoff, H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, *5*, 111–125.

Franz, M., Schölkopf, B., Mallot, H., & Bülthoff, H. (1998). Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, *79*, 191–202.

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory: Eurocolt*, *95*, 23–37 Springer Berlin/Heidelberg.

Fukushi, T., & Wehner, R. (2004). Navigation in wood ants *Formica japonica*: Context dependent use of landmarks. *Journal of Experimental Biology*, *207*(19), 3431–3439.

Gabor, D. (1946). Theory of communication. *Journal of the Institute of Electrical Engineers*, *93*(3), 429–457.

Gaussier, P., & Zrehen, S. (1995). Perac: A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, *16*(2-4), 291–320.

Giovannangeli, C., Gaussier, P., & Désilles, G. (2006). Robust mapless outdoor vision-based navigation. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, (pp. 3293–3300), 9–15 Oct. 2006.

Graham, P., & Cheng, K. (2009). Ants use the panoramic skyline as a visual cue during navigation. *Current Biology*, *19*(20), R935–R937.

Graham, P., & Collett, T. S. (2002). View-based navigation in insects: How wood ants (*Formica rufa* L.) look at and are guided by extended landmarks. *Journal of Experimental Biology*, *205*(16), 2499–2509.

Graham, P., Philippides, A., & Baddeley, B. (2010). Animal cognition: Multi-modal interactions in ant learning. *Current Biology*, *20*, R639–R640.

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, *69*(3), 331–371.

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), 1–12.

Papageorgiou, C., Oren, M., & Poggio, T. (1998). A general framework for object detection. In *Sixth International Conference on Computer Vision (ICCV'98)*, IEEE Computer Society, Los Alamitos, CA, USA, p. 555).

Philippides, A., Baddeley, B., Cheng, K., & Graham, P. (2011). How might ants use panoramic views for route navigation?. *Journal of Experimental Biology* (in press).

Rosengren, R., & Fortelius, W. (1986). Ortstreue in foraging ants of the *Formica rufa* group – hierarchy of orienting cues and long-term memory. *Insectes Sociaux*, *33*(3), 306–337.

Smith, L., Philippides, A., Graham, P., Baddeley, B., & Husbands, P. (2007). Linked local navigation for visual route guidance. *Adaptive Behavior*, *15*(3), 257–271.

Smith, L., Philippides, A., Graham, P., & Husbands, P. (2008). Linked local visual navigation and robustness to motor noise and route displacement. In *Proceedings of the 10th International Conference on Simulation of Adaptive Behavior: from Animals to Animats* (ed. M. Asada, J. C. Hallam, J-A. Meyer and J. Tani). Lecture Notes in Computer Science 5040, pp. 179–188. Berlin, Heidelberg: Springer-Verlag.

Stürzl, W., & Zeil, J. (2007). Depth, contrast and view-based homing in outdoor scenes. *Biological Cybernetics*, *96*, 519–531.

Vardy, A. (2006). Long-range visual homing. In *IEEE International Conference on Robotics and Biomimetics* (pp. 220–226).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 1, p. 511–518).

Wehner, R. (1996). Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, *199*(1), 129–140.

Wehner, R., Boyer, M., Loertscher, F., Sommer, S., & Menzi, U. (2006). Ant navigation: One-way routes rather than maps. *Current Biology*, *16*(1), 75–79.

Wehner, R., & Raber, F. (1979). Visual spatial memory in desert ants, *Cataglyphis bicolor* (Hymenoptera: Formicidae). *Cellular and Molecular Life Sciences (CMLS)*, *35*(12), 1569–1571.

Zeil, J., Hofmann, M., & Chahl, J. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A*, *20*(3), 450–469.

About the Authors

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*: bartbaddeley@googlemail.com

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. *Address*: Centre for Computational Neuroscience and Robotics, School of Life Sciences, University of Sussex, Brighton, UK. *E-mail*: paulgr@sussex.ac.uk

Andrew Philippides is a lecturer in the CCNR at the University of Sussex. He is interested in the modeling of biological systems at both behavioral and neuronal levels. *Address*: Centre for Computational Neuroscience and Robotics, School of Informatics, University of Sussex, Brighton, UK. *E-mail*: andrewop@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. *Address*: Centre for Computational Neuroscience and Robotics, School of Informatics, University of Sussex, Brighton, UK. *E-mail*: p.husbands@sussex.ac.uk