

A Generalized Multi-Snapshot Model for 3D Homing and Route Following

Adaptive Behavior
XX(X):1–15
©The Author(s) 2020
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Dario Differt¹ and Wolfgang Stürzl²

Abstract

Inspired by the learning walks of the ant *Ocymyrmex robustior*, the original multi-snapshot model was introduced, which – in contrast to the classical “single snapshot at the goal” model – collects multiple snapshots in the vicinity of the goal location that subsequently can be used for homing, i.e. for guiding the return to the goal. In this study we show that the multi-snapshot model can be generalized to homing in three dimensions. In addition to capturing snapshots at positions shifted in all three dimensions, we suggest to decouple the home direction from the orientation of snapshots and to associate a home vector with each snapshot. We then propose a modification of the multi-snapshot model for three-dimensional route following and evaluate its performance in an accurate reconstruction of a real environment. As an illumination-invariant alternative to greyscale images we also examine sky-segmented images. We use spherical harmonics as efficient representation of panoramic images enabling low memory usage and fast similarity estimation of rotated images. The results show that our approach can steer an agent reliably along a route, making it also suitable for robotic applications using on-board computers with limited resources.

Keywords

Visual navigation, Route Following, Flying Insects, View-based Homing, Spherical Harmonics

1 Introduction

As has been shown in numerous studies, insects combine multiple strategies as, for example, path integration, odor tracking, and visual guidance to navigate between feeding sites and their nests (reviews: [Graham 2010](#); [Srinivasan 2011](#); [Wolf 2011](#); [Wystrach and Graham 2012](#); [Zeil 2012](#)). A seminal model for visual navigation in insects is the classical snapshot model ([Cartwright and Collett 1983](#)) which states that insects collect snapshots – basically retinotopic images or “views” – at places of interest, e.g. at the nest or a food source.

Homing with single snapshot For returning to the place where the snapshot was taken (“homing”) a number of approaches have been proposed. An example is the gradient method proposed by [Zeil et al. \(2003\)](#), which is based on the finding that image similarity increases quite smoothly with decreasing distance to goal, i.e. the closer two positions are in space the higher is the similarity of the corresponding images. It continuously estimates the goal direction from the local gradient of the image similarity to the goal view, which can be approximated by capturing images in two (or three for homing in 3D) orthogonal directions in addition to the current view. Instead of collecting additional views, image warping methods explicitly ([Franz et al. 1998b](#)) or implicitly ([Möller 2009](#); [Möller et al. 2010, 2014](#)) compute or generate “hypothetical views” from the current view by distorting (=“warping”) it for a number of possible movements based on simplifying assumptions about the depth structure of the scene and output the direction for which the highest similarity to the snapshot could be achieved. They thus avoid probing translational movements but have higher computational complexity. In summary,

single snapshot homing methods estimate the goal direction by either computing or actively collecting views around the current position. It should be mentioned that there are also methods that estimate home vectors (or relative poses) from corresponding features in current view and the goal snapshot based on optic flow, local intensity patches or feature descriptors ([Vardy and Möller 2005](#); [Fleer and Möller 2017](#)). As an in depth-discussion of these methods is beyond the scope of this paper we just note that they are computationally more complex than the multi-snapshot model considered in this study.

Homing with multiple snapshots An alternative to elaborating the homing procedure is to simplify homing by investing more resources during learning. An elegant example is the local homing model by [Graham et al. \(2010\)](#) that, instead of a single snapshot at the goal position, uses multiple snapshots around the goal. To navigate back to the goal location, the agent only has to compare its current view with all stored snapshots and to choose the home direction associated with the most similar snapshot.

It is interesting to note the strong relationship between the multi-snapshot model and the single snapshot homing methods, as illustrated in [Fig. 1](#): If, instead of capturing

¹Computer Engineering Group, Faculty of Technology, Bielefeld University, Germany

²Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

Corresponding author:

Wolfgang Stürzl, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Muenchner Str. 20, 82234 Wessling, Germany
Email: wolfgang.stuerzl@dlr.de

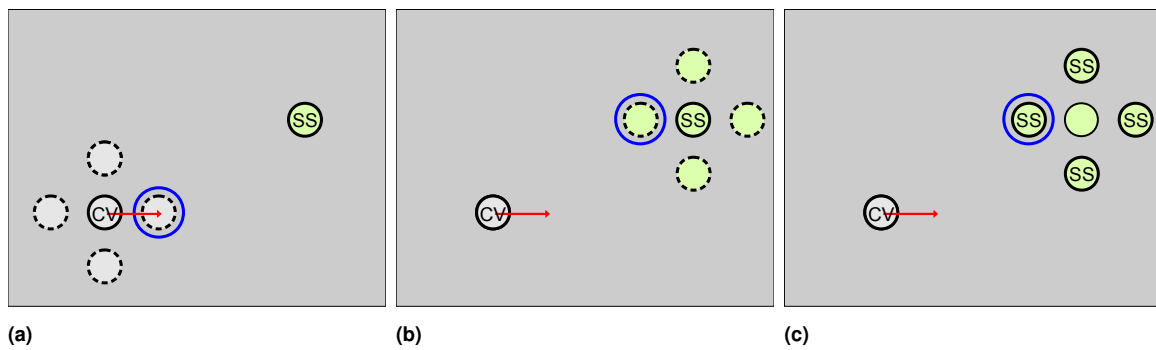


Figure 1. Illustration of the relationship between single-snapshot homing methods and the multi-snapshot model. (a) Single snapshot methods use additional views (dashed circles) around the current position by either direct sampling, i.e. by moving to neighboring positions for recording images, or by computing hypothetical views generated using image warping. In the next step the agent would, as depicted by the red arrow, move to the position of the view (highlighted by blue circle) that had highest similarity to the snapshot. (b) Alternatively, views around the goal position could be used, e.g. hypothetical views computed from the goal snapshot with the warping method (basically swapping snapshot and current view in the usual warping approach). In this case the agent has to move in the direction opposite to the direction of the most similar view with respect to the goal, which is equal to the direction of the goal as seen from the position of the best-matching view. (c) Instead of computing hypothetical views, the multi-snapshot model assumes that snapshots are captured around the goal position (but not necessarily at the goal). As in (b) the agent would move in the direction of the vector pointing from the position of the best-matching snapshot to the goal.

additional images around the current position, an agent would have supplementary images around the goal, it could select the one most similar to the current view and move in the direction opposite to the direction of the most similar image with respect to the goal. Capturing additional images around the goal position is of course not possible during homing but generating hypothetical views from the goal snapshot would work for the warping model. As the goal snapshot does not change, the additional images could also be pre-computed or – as the multi-snapshot model proposes – simply captured immediately after departure from the goal resulting in minimal computational load but increased memory usage.

Route navigation All local homing methods have in common that they are limited to a certain area around the goal, called the “catchment area”, as, for instance, due to occlusions, the current view will not share enough features with a snapshot for larger distances from the goal. Navigation over longer distances can be achieved by collecting a sequence of snapshots along a route that then can be repeated from start to end. [Baddeley et al. \(2012\)](#) use rotations on the spot that are motivated by scanning movements observed in ants ([Wystrach et al. 2014](#); [Zeil et al. 2014](#)) to align current views with the route. However translational deviations may occur that can increase with time, which could be avoided by employing one of the local homing methods mentioned before with the currently most similar route snapshot as goal snapshot. Taking this into account, an extension to the route navigation model of ([Baddeley et al. 2012](#)) was proposed in ([Gaffin et al. 2015](#); [Gaffin and Brayfield 2016](#)) that, similar to the gradient method in ([Zeil et al. 2003](#)), uses translational scanning movements, i.e. additional views around the current position are captured and compared to the stored snapshots allowing to compensate for drift. In contrast, our approach proposes, inspired by the multi-snapshot model, to simplify reliable route following by elaborating and refining the route learning procedure instead. In addition, it is, to the best of our knowledge, the only insect-inspired 3D route navigation method. While this is not important for ants and other

species that walk on the ground it matters for flying insects. Recently, ([J. Müller et al. 2018](#)) adapted an insect mushroom body model ([Ardin et al. 2016](#)), which essentially realizes a biologically plausible version of the view familiarity classifier proposed by ([Baddeley et al. 2012](#)), for honeybee foraging flights, but did not control height above ground as altitude remained fixed in the simulation.

In the following we first recapitulate the multi-snapshot model and then present modifications and generalizations for homing in 3D (section 2) that form the basis for an efficient and robust route following method. The approach and its implementation will be described in detail in sections 3 & 4. We then present in section 5 results of several 3D route navigation experiments in a simulated environment. Finally, we discuss our approach and suggest ideas for future work (section 6).

2 The Multi-Snapshot Model

While the idea of collecting a set of snapshots for homing has been described earlier, e.g. in ([Judd and Collett 1998](#); [Nicholson et al. 1999](#)) for ants and in ([Cartwright and Collett 1987](#)) for bees, we will focus on the multi-snapshot model originally presented in ([Graham et al. 2010](#)) and later more extensively discussed in ([Narendra et al. 2013](#); [Dewar et al. 2014](#)). It is based on observations of learning walks of the ant *Ocymyrmex robustior* ([Müller and Wehner 2010](#); for a recent comparison of learning walks in different ant species see [Zeil and Fleischmann 2019](#)). After leaving the nest, the ant walks around the nest entrance and stops at various locations and turns towards the nest location as sketched in the upper drawing of Fig. 2 a. Following the idea of the snapshot model ([Cartwright and Collett 1983](#)), it is assumed that the ant stores a snapshot, i.e. the view of the current visual scenery, each time it is looking towards the nest. By doing so, a set of snapshots from various locations around the nest is collected. At return, the ant, according to the multi-snapshot model ([Graham et al. 2010](#)), can move towards the nest by using solely the stored snapshots as

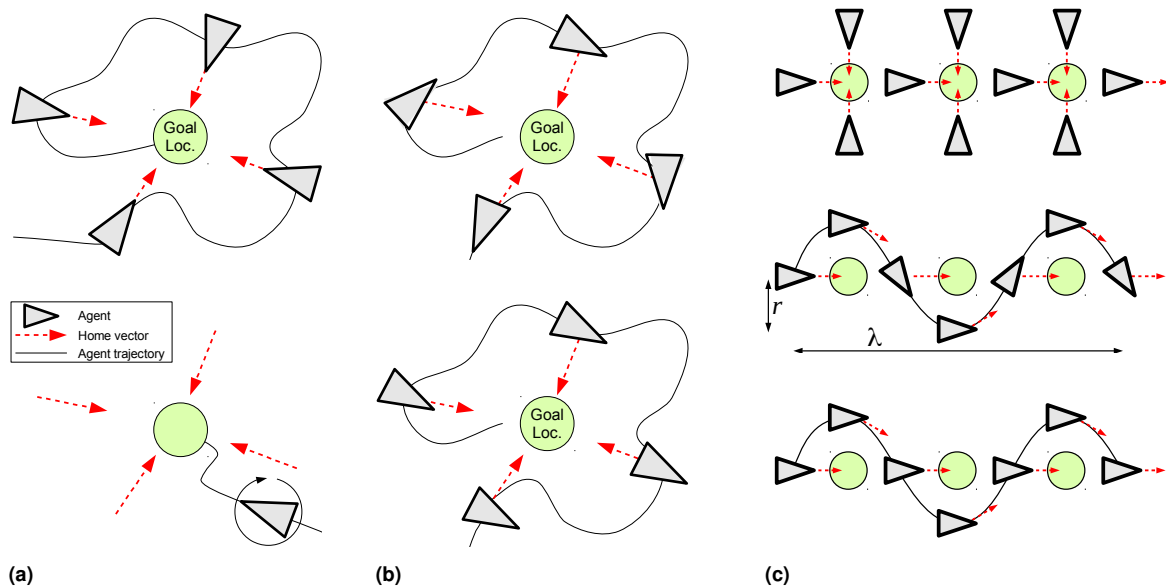


Figure 2. Illustration (in 2D) of the multi-snapshot model to perform homing (a), the proposed generalizations (b) and our approach to use it for route following (c). (a) The original multi-snapshot model: In a first step, the agent collects a set of snapshots facing the goal location (top). The collected snapshots are later used to steer the agent back towards the goal location by always moving in the direction at which the most similar snapshot was captured (bottom). Since the agent can be oriented arbitrary, the agent needs to rotate at the spot (depicted by circular arrow) to compare its current view with all captured snapshots. (b) Extension of the multi-snapshot model: Besides arranging snapshots in 3D, we propose that the vectors associated with snapshots can be independent of snapshot orientation (top). In addition, also movement direction and snapshot orientation can be decoupled, which can reduce the range of rotations to be tested (bottom). (c) Route following can be performed by concatenating multiple homing tasks (top). Our approach decouples the home vector direction and the agent’s orientation, which allows us to propose a simple trajectory to capture snapshots. For instance, snapshots can be either aligned with the trajectory during learning (middle) or with the direction of the route (bottom).

shown in lower part of Fig. 2a: While turning on the spot, a behavior which has been observed in a number of ant species (Wystrach et al. 2014; Zeil et al. 2014), the ant continuously compares the current view with all collected snapshots. By walking in the direction for which the best match could be found, the ant gets closer to the nest. The key idea of the multi-snapshot model is that the collected snapshots look different depending on the location of the agent relative to the goal location. In more detail, the stored snapshots contain implicit information about the orientation of the agent as well as the relative distance to objects around the goal location and hence about the movement direction. By comparing the current view with all snapshots, one eventually finds a snapshot which was captured in the vicinity of the current location. In contrast, other homing methods, e.g. min-warping (Möller et al. 2010), only use a single snapshot collected at the goal location such that the relative orientation and the movement direction of the agent have to be estimated from the current view and the snapshot alone. In the multi-snapshot model, however, the complexity of the problem is reduced due to the movements of the agent, in particular during the learning phase when snapshots at different positions are collected, such that the process of homing collapses to a number of computationally cheap image comparisons. Also the catchment area, i.e. the area where the agent can be visually guided towards the goal, can be easily extended by simply increasing the number of snapshots.

2.1 Extension to 3D

Until now, the multi-snapshot model has only been used for two-dimensional navigation – maybe with the exception of the homing model for ground-nesting wasps (Stürzl et al. 2016), see discussion. In the following we introduce the multi-snapshot model for three-dimensional movements, which makes it particularly interesting for flying agents. Let us denote the current view – commonly represented by a panoramic image – by some feature vector $\vec{CV} \in \mathbb{R}^n$ and the set of N snapshots by feature vectors $\vec{SS}_i \in \mathbb{R}^n$, $i = 1, 2, \dots, N$. Note that we do not explicitly describe the current view and the snapshots for example by pixel matrices, which would be a suitable choice to represent panoramic images, but instead by an arbitrary feature vector. This generalization has been made to allow alternative representations of panoramic images, e.g. by Fourier coefficient vectors as used in our implementation.

For arbitrary three-dimensional movement, the orientation of the agent at its current location can be represented by a rotation matrix $\mathbf{R} \in \text{SO}(3)$, where $\text{SO}(3)$ is the set of all three-dimensional rotation matrices. In the following we denote by $\mathbf{R} \circ \vec{CV}$ and $\mathbf{R} \circ \vec{SS}_i$ a rotation applied to the current view \vec{CV} and snapshot \vec{SS}_i , respectively. The operation ‘ \circ ’ depends on the representation of the panoramic images. Rotations can either be performed by movement of the agent, e.g. by an ant rotating on the spot, or by simulating the rotations internally.

For rotations around the vertical axis, the visual compass as suggested by Zeil et al. (2003) can be used to estimate relative orientation between images. For

arbitrary orientations, as assumed throughout this study, we use a three-dimensional implementation of the visual compass (Differt 2017b). Finally, let $\|\cdot\|$ be some image distance function. By searching for the rotation which minimizes the image distance between the current view and all snapshots,

$$\mathbf{R}^* = \arg \min_{\mathbf{R} \in \text{SO}(3)} \left(\min_i \|\mathbf{R} \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\| \right), \quad (1)$$

we obtain the unit vector pointing in the home direction as $\vec{H}\vec{V} = \mathbf{R}^* \vec{V}_c \in \mathbb{R}^3$, where \vec{V}_c defines the current movement direction. Here, we implicitly assume that the orientation of a view or snapshot has a fixed relation to the movement or home direction, e.g. the center of the camera image is aligned with the translation direction.

2.2 Decoupling of home direction and view orientation

Snapshot orientation does not necessarily have to be aligned with home direction. Instead, a direction vector can be associated to each snapshot, as illustrated in Fig. 2 b. The home direction can then be estimated according to

$$\vec{H}\vec{V} = \mathbf{R}_{i^*}^* \vec{H}\vec{V}_{i^*}, \quad (2)$$

$$i^* = \arg \min_i \|\mathbf{R}_i^* \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\|, \quad (3)$$

$$\mathbf{R}_i^* = \arg \min_{\mathbf{R} \in \text{SO}(3)} \|\mathbf{R} \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\|, \quad (4)$$

where we assume that each associated direction vector $\vec{H}\vec{V}_i$ is given in the coordinate frame of snapshot i and that the home direction $\vec{H}\vec{V}$ is estimated in the frame of the current view.

As illustrated in the upper drawing of Fig. 2 b, snapshots may be aligned with movement direction to allow for fast capturing of snapshots, and the associated direction vectors point towards the goal, which is in a different direction for each snapshot. However, snapshot orientation can also be independent of the movement direction. For instance, in flying insects there is no fixed relation between head orientation and flight direction. A nearly constant orientation of snapshots, as depicted in the lower part of Fig. 2 b, can simplify similarity estimation between the current view and the stored snapshots during homing.

Note that in addition to the home direction also a distance as well as an orientation difference could be associated with each snapshot.

Weighted vector sum Selecting the home direction associated with the best-matching snapshot (“arg min selection”), may lead to strong changes in direction, in particular, if only a few snapshots have been captured. Instead of Eqs. (2) and (3), a weighted vector sum over all or a subset of vectors can be used,

$$\vec{H}\vec{V} = \sum_{i=1}^N w_i \mathbf{R}_i^* \vec{H}\vec{V}_i, \quad (5)$$

where weights w_i should decrease with image distance between snapshot $\vec{S}\vec{S}_i$ and current view $\vec{C}\vec{V}$.

A weighted average (of snapshot aligned home directions) was already proposed in (Graham et al. 2010), but no

mathematical definition of the weighting function given. In (Dewar et al. 2014) weights were chosen according to, in our notation, $w_i = \frac{\min_k \|\mathbf{R}_i^* \circ \vec{C}\vec{V} - \vec{S}\vec{S}_k\|}{\|\mathbf{R}_i^* \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\|}$. Instead, we propose to use the soft(arg)min function (Bishop 2006), i.e.

$$w_i = \frac{\hat{w}_i}{\sum_{k=1}^N \hat{w}_k}, \quad \hat{w}_i = e^{-\beta_i \|\mathbf{R}_i^* \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\|}, \quad (6)$$

as it provides, depending on the choice of $\beta_i \in \mathbb{R}_{\geq 0}$, a smooth transition between the weighted average and the arg min selection. Note that for $\beta_i = \beta \forall i$ the limit $\beta \rightarrow \infty$ leads to $w_i \rightarrow \delta_{ii^*}$ (where δ_{ik} is the Kronecker delta, i.e. $\delta_{ik} = 1$ if $i = k$, and 0 otherwise) making (5) identical to (2) and (3).

Close to the goal, the length of the estimated home vector $\vec{H}\vec{V}$ should become smaller and ideally vanish at the goal position. In case of snapshot positions being evenly distributed around the goal this can be achieved by setting

$$\beta_i = \alpha \|\mathbf{R}_{i0} \circ \vec{S}\vec{S}_0 - \vec{S}\vec{S}_i\|^{-1}, \quad (7)$$

where $\vec{S}\vec{S}_0$ represents an additional snapshot captured at the goal position and \mathbf{R}_{i0} the orientation difference of $\vec{S}\vec{S}_0$ with respect to $\vec{S}\vec{S}_i$, and $\alpha \in \mathbb{R}_{\geq 0}$ is a parameter (independent of i). Note that just the $\{\beta_i\}$ but not $\vec{S}\vec{S}_0$ have to be stored. If the current position during return is very close to the goal position (and therefore $\vec{C}\vec{V}$ very similar to $\vec{S}\vec{S}_0$) we have

$$\alpha \frac{\|\mathbf{R}_i^* \circ \vec{C}\vec{V} - \vec{S}\vec{S}_i\|}{\|\mathbf{R}_{i0} \circ \vec{S}\vec{S}_0 - \vec{S}\vec{S}_i\|} \approx \alpha \frac{\|\mathbf{R}_{i0} \circ \vec{S}\vec{S}_0 - \vec{S}\vec{S}_i\|}{\|\mathbf{R}_{i0} \circ \vec{S}\vec{S}_0 - \vec{S}\vec{S}_i\|} = \alpha,$$

and therefore $w_i \approx N^{-1} \forall i$ resulting in

$$\vec{H}\vec{V} \approx N^{-1} \sum_{i=1}^N \mathbf{R}_i^* \vec{H}\vec{V}_i \approx \vec{0}.$$

Examples of 3D home vectors computed using the direction vector associated with the most similar snapshot, Eqs. (2),(3), or the weighted vector sum, Eqs. (5),(6) are shown in Fig. 3. Two different views of the results for just four snapshots, the minimum number of snapshots for homing in 3D, in a tetraeder-like arrangement are provided in Fig. 3 a and Fig. 3 b. For Fig. 3 c, six snapshots were used where pairs of snapshots lie on orthogonal axes. The weighted sum leads to a smaller angular error between the estimated and the true home direction. For the four (six) snapshots, the average angular error is reduced to 26° (17°) compared to 43° (40°) when selecting the direction associated with the most similar snapshot. In this example all snapshots and “current views” were rendered with the same orientation so that the search for the best orientation was omitted (see Fig. 3 e for an example snapshot at the goal position). Image distances were calculated as the sum of squared pixel differences.

3 3D Route Following

In this study we are especially interested in visually guided 3D route following by means of previously collected snapshots. Therefore we adapt the multi-snapshot model, which was discussed in the previous section in the context of homing towards a goal location, for this task.

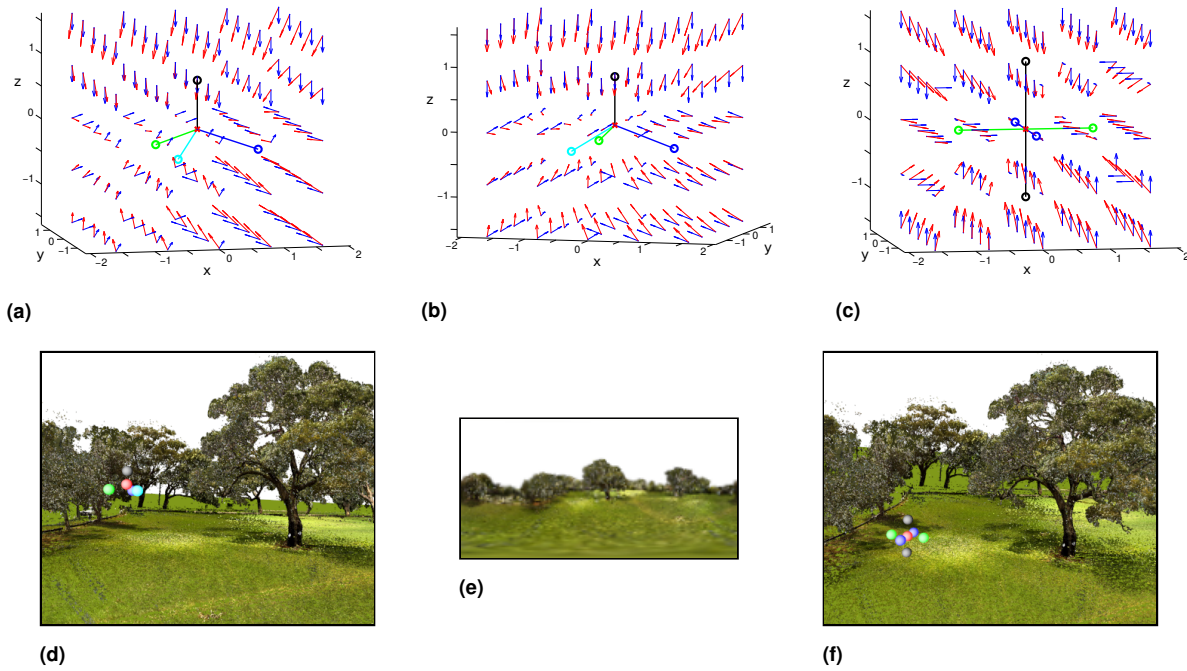


Figure 3. Examples of home vectors computed with the 3D multi-snapshot model with four snapshots (a), (b) or six snapshots (c) around the same goal position approx. 5 m above ground level. (a) and (b) give two different views of the home vectors for the tetraeder-like arrangement of snapshot positions. The snapshot positions are indicated by circles of different colour. Blue vectors were calculated using the direction associated with the most similar snapshot, Eqs. (2),(3). Red vectors were computed using the weighted vector sum, Eqs. (5) and (6); an additional snapshot at the goal was used for determining the β_i according to Eq. (7) with parameter α set to 0.1. Home vectors were calculated on a $5 \times 5 \times 5$ grid with step size of 0.5 m and the goal position in the origin. Axes units are in meters. (d) Rendered overview for the four snapshot configuration (snapshot positions marked by green, blue, light blue and grey spheres, goal position by red sphere). (e) Panoramic image at the goal rendered with 180×90 pixels. The large tree visible on the right of the rendered overviews in (d) and (f) is in the center of the image. (f) Rendered overview for the six snapshot configuration with same goal position (red sphere) as in (d).

Route following can be separated into different sub-tasks: (1) As a first step that interestingly is skipped in many insect-inspired approaches (see discussion), an agent may localize itself on the route in order to reduce the search space and the chance of mismatches. (2) Most importantly, the steering of the agent needs to be adjusted such that it follows the direction of the route. (3) Furthermore, due to disturbances the agent may leave the route and the steering needs to be adjusted in this case to lead the agent back onto it. Without compensating offsets from the route the agent might drift away, especially on long routes.

A straight-forward way is to implement route following as a sequence of single homing tasks as it is shown in the upper drawing of Fig. 2 c. Snapshots are collected around subsequent goal locations along a route. Then route following can be performed as “linked local navigation” (Smith et al. 2007) by homing towards the first goal location and, as soon as the agent arrives, homing towards the next goal location until the agent reaches the end of the route. However, using the multi-snapshot model as introduced in the previous section, we would need to collect snapshots from all directions around each goal location. As can be inferred from the topmost drawing in Fig. 2 c, there is no simple trajectory the agent could follow to collect the required snapshots.

To overcome these problems, we use the following approach: First, as already introduced in section 2.2, we do not require the agent to face towards the goal location when capturing snapshots. Instead, the agent can be arbitrarily

oriented and a vector is associated to each snapshot, as shown in the drawings in the middle and bottom row of Fig. 2 c. Second, we do not capture snapshots from multiple directions, since this would require the agent to move along a complex trajectory. Instead, we move the agent along a sine-wave trajectory as illustrated in the lower parts of Fig. 2 c.

For three-dimensional movement, the sine-wave can be generalized as a helix. Both the helix trajectory as well as the snapshot collection is sketched in Fig. 4 a. As we will show by means of simulation in the result section, the helix trajectory can be used to perform route following in three-dimensional space. Additionally, we propose a curved zigzag trajectory which only requires the agent to oscillate around the route, see Fig. 4 b, as this type of trajectory is likely to be a better approximation of what can be observed in flying insects.

A two-dimensional sketch of the vector fields resulting from our approach can be seen in Fig. 4 c for two helices with same wavelength, $\lambda = 2$ m, but different radii, $r = 0.5$ m (top) and $r = 1$ m (bottom). The vector field is partitioned into regions around each snapshot that have minimal distance to the respective snapshot position and point in the direction of the direction associated with this closest snapshot. Note that we used the spatial distance here as an approximation to the image distance, i.e. to the arg min selection of the snapshot most similar to the view at each position, Eqs. (2),(3). As can be seen, the home vector field steers the agent inside the helix, however it does not converge exactly to the route (i.e. the center of the helix) but to a

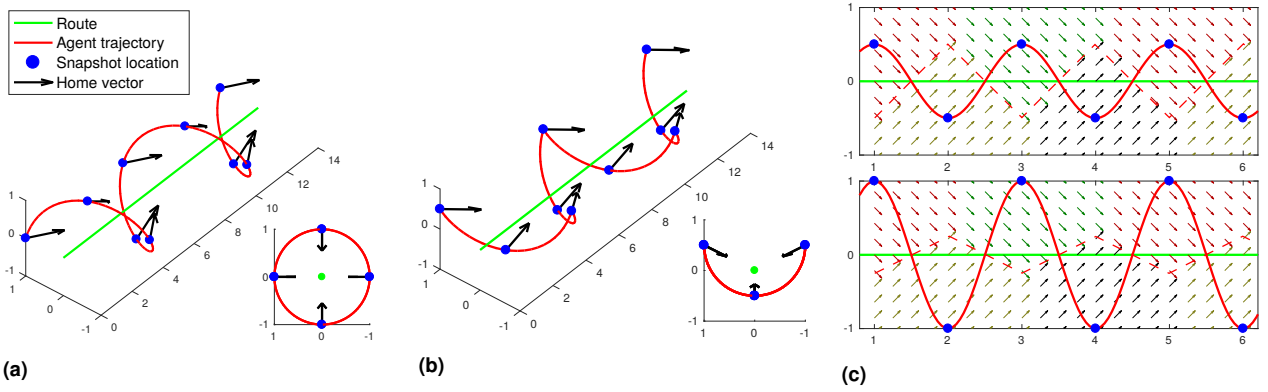


Figure 4. Illustration of the types of trajectories for collecting snapshots around the ideal route that are proposed and tested in this work. Shown are 3D and axial views of a helix (a) and a curved zigzag trajectory (b) with radius r and length λ . (c) 2D sketches of exemplary vector fields for $r = \frac{1}{4}\lambda$ (top) and $r = \frac{1}{2}\lambda$ (bottom) assuming the agent always chooses the vector associated with the closest snapshot. The approximate area in which the agent is closest to a snapshot is indicated by home vectors of the same colour. The vector fields do not converge to the route directly but to curves of triangular shape (dashed red lines) with vertical extent proportional to λ/r .

triangular curve of same wavelength and opposite phase. The amplitude of the deviation scales with the ratio of wavelength and radius, i.e. the further away the snapshots are along the route for a given radius the larger the expected deviations are. A weighted average, Eq. (5), could be employed in order to reduce such deviations from the ideal route. However, the effect of the weighted average is pronounced only if vectors associated to neighboring snapshots differ by large angles, as for instance in the examples sketched in Fig. 4c, or in the 3D homing examples shown in Fig. 3. For the route following experiments presented in this study, neighboring vectors differ only by about 20° . For the sake of simplicity we will therefore use the arg min selection for route following.

4 Methods

4.1 3D Simulation

To render panoramic images, we use a 3D model of an environment, which was recorded in an urban park in Canberra, Australia, using a laser-scanner with color camera attached (Stürzl et al. 2015). These panoramic images can be rendered at arbitrary locations and orientations inside the simulated environment, allowing us to simulate the visual surrounding of a freely moving agent in a three-dimensional environment. Rendered panoramic images have a resolution of 180×90 pixels and are converted to floating-point greyscale images (values ranging between 0 and 1). As an illumination-robust alternative, we also create sky-segmented images (Differt and Möller 2016), in which all pixels of the sky are white and all pixels of ground objects are black, by applying a simple threshold of 0.9 to greyscale image. Examples of a rendered greyscale and sky-segmented image can be found in Fig. 5.

4.2 Image Distance using Spherical Harmonics

In this section we describe in more detail our proposed method as we used it for the tests in the results section. Our implementation is based on the C++ library libShc (Differt 2017a) to represent panoramic images in frequency domain.

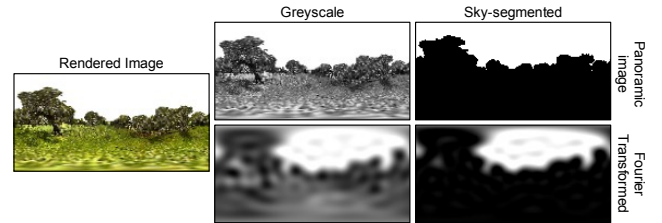


Figure 5. Rendered images are either converted to greyscale or to sky-segmented images for our proposed route following method. Images are represented by a Fourier coefficient vector; the inverse Fourier transformed images for $l = 15$ frequencies are shown in the bottom row.

We Fourier transform the panoramic images by projecting them into the basis of real spherical harmonics. For this task we use 10^4 equally distributed sampling points regarding the topology of the unit sphere. We only use the first $l = 15$ frequencies such that we can represent each panoramic image by a Fourier coefficient vector with a total of $l^2 = 225$ real-valued entries. The Fourier coefficient vectors are then used to represent current views and snapshots. As is illustrated in the bottom row of Fig. 5 by means of inverse Fourier transformation, the 225 spherical Fourier coefficients contain the information of low-pass filtered images. In the following we will continue to use the terms “greyscale” and “sky-segmented images” even if the route following was actually implemented using spherical harmonics. The reader can always think of low-pass filtered images with same results but less efficient computation.

To compare two different panoramic images $\vec{C}\vec{V}$ and $\vec{S}\vec{S}$ represented by their Fourier coefficient vectors in frequency domain, we use the integral squared error (ISE). The ISE can be computed directly in the basis of real spherical harmonics as

$$\text{ISE}(\vec{C}\vec{V}, \vec{S}\vec{S}) = \|\vec{C}\vec{V} - \vec{S}\vec{S}\|^2. \quad (8)$$

Note that the ISE defined in Eq. (8) is (up to a constant factor) mathematically equivalent to the sum of squared pixel differences of low-pass filtered images (since we limit frequencies to $l \leq 15$). However, rotations around

arbitrary axes can be realized more efficiently using spherical harmonics.

To rotationally align the current view with the collected snapshots we use the visual 3D compass described in (Differt 2017b). The visual 3D compass uses a Tait-Bryan parameterization (yaw-pitch-roll) for three-dimensional rotations to systematically search for the rotation which minimizes the ISE between the current view and each snapshot. This search is implemented as a coarse-to-fine search and starts to search in 16° steps around all axes. The pitch and roll rotations are limited to $\pm 32^\circ$ since we assume that the tilt of the agent is limited. Afterwards, the search is refined in 8° , 4° , and 2° steps around the found minima. In addition, we limited the yaw angle to $\pm 48^\circ$, since we assume that the agent moves smoothly.

4.3 Implementation of Route Navigation

Snapshots are collected in steps of 1 m along the route. More precisely, the spacing between the projections of snapshot positions onto the route is 1 m, which means that distance between successive snapshot positions for the helix or the curved zigzag learning trajectory (with $r > 0$) is larger than 1 m and increases with radius. The wavelength is set to $\lambda = 4$ m. The curved zigzag trajectory is created from the helix by mirroring points above the route at the horizontal plane and moving the resulting trajectory upwards by $1/2 r$ to center it on the route (see also Fig. 4 a and b).

A home vector is associated with each snapshot to steer the agent during route following. For the helix and curved zigzag trajectories, the home vectors should point towards the route to avoid drift as well as along the route to move the agent forward. Let us denote by $\vec{u}_i \in \mathbb{R}^3$ the orthogonal projection of the location of snapshot i onto the route and by $\vec{v}_i \in \mathbb{R}^3$ the vector tangential to the route (note that these vectors are orthogonal), then we define the home vector as

$$\vec{H}\vec{V}_i = \alpha \vec{u}_i + \vec{v}_i, \quad (9)$$

with $\alpha < 0$. We set $\alpha = -1/4$ in all experiments but the exact value is not critical. In cases where the learning trajectory is identical to the route and therefore $\vec{u}_i = \vec{0}$, the home vector is tangential to the route, $\vec{H}\vec{V}_i = \vec{v}_i$.

During route following, the simulated agent moves in discrete 0.3 m steps. At the start location it is oriented tangentially to the route. Afterwards, the agent is always oriented in the direction of the current home vector, i.e. the home vector associated with the snapshot that according to the ISE, Eq. (8), is most similar to the current view. For selecting the best-matching snapshot, only the previous and upcoming 20 snapshots (i.e. around ± 20 m) relative to the previous best-matching snapshot are considered, which has two advantages: (1) the computation time is independent of the total number of snapshots captured along the route, (2) by only using snapshots which are relative close to the agents position, the probability of incorrect matches is decreased.

To simulate disturbances, we add noise to the home vectors stored at each snapshot locations such that the angular difference $\Delta\phi \in \mathbb{R}_{\geq 0}$ between the correct and noisy home vectors follows a half-normal distribution $p(\Delta\phi; \sigma) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma} e^{-\frac{\Delta\phi^2}{2\sigma^2}}$ with $\langle \Delta\phi^2 \rangle = \sigma^2$.

A route following run is considered successful if the agent arrives within 2 m of the last snapshot location on the route.

5 Experiments

To show the feasibility of our route navigation method, we performed two different types of experiments in the simulated outdoor environment. In the first set of experiments (“Proof of Concept”) we demonstrate the importance of collecting snapshots not just along but also around the route. The second set (“Systematic Analysis”) examines the influence of noise, varying helix radii, and translational offset from the route.

5.1 Proof of Concept

For the initial experiment a 200 m long route in the central area of the simulated environment was selected, see Fig. 6 a, and, as described in section 4.3, snapshots were collected in steps of 1 m directly on the route, which in this case is identically to the learning trajectory (helix or curved zigzag trajectory of radius $r = 0$ m). As can be seen in Fig. 6 b, with perfect home vectors (no noise added) no significant errors occur. Small deviations, that might increase for longer routes, are mainly due to the discrete step size and sampling of snapshots. Major errors will only occur in the rare case that a wrong snapshot is chosen, i.e. the ISE to (the rotationally aligned) current view for that snapshot is smaller than for the snapshot of the closest positions on the learning trajectory. However, with noisy home vectors ($\sigma = 10^\circ$) significant deviations from the route can occur increasing the chance of mismatches that can lead to complete failure. Fig. 6 c shows such a case for route navigation with sky-segmented images. Small deviations from the learning trajectory in combination with close objects (trees in this case) can lead to mismatches, i.e. the wrong snapshot and, hence, the wrong vector is selected. Mismatches leading to complete failures were less likely for greyscale images but can be found as well, as illustrated in the example presented in Fig. 6 d,e.

In contrast, for the proposed route navigation using a helix or curved zigzag trajectory (with radius $r = 1$ m) around the route as learning trajectory for collecting snapshots and home vectors no complete failure was observed even for noisy home vectors, Fig. 7. We also examined our approach on another route (route “high”, Fig. 7 b) that was basically identical to the initial route (route “low”, Fig. 7 a) except that, after around 50 m instead of bypassing a tree it passes over it. After around 100 m both routes merge again. As illustrated in Fig. 7 c, both routes are mainly around 5 m to 10 m above ground level. However, while passing over the tree, route “high” reaches a height of about 16 m.

All presented route following tests (with $r = 1$ m) were successful, indicating that our approach, despite low requirements on computational power and memory, enables reliable route following even on longer routes, as well as in environments with complex depth structures, see images in figure 7 e. This includes open spaces (A), high altitudes (B), and narrow passages between trees (C). However using greyscale images seemed to be more reliable than using sky-segmented images. Especially close to the highlighted locations (B) and (C), route following using sky-segmented images sometimes nearly failed, see green curve



(a) Rendered views of the simulated environment

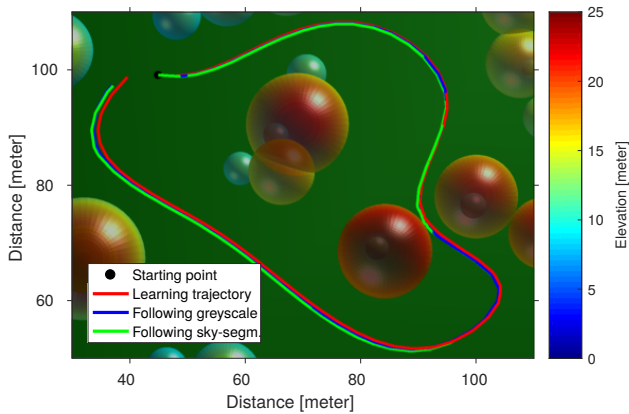
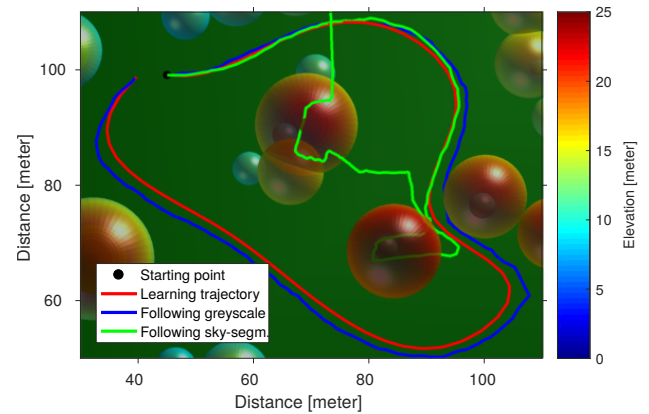
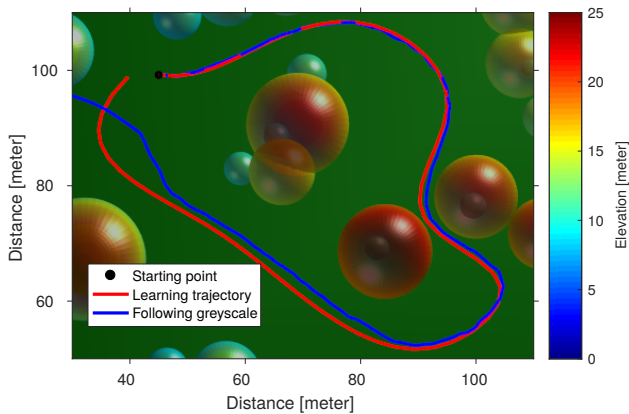
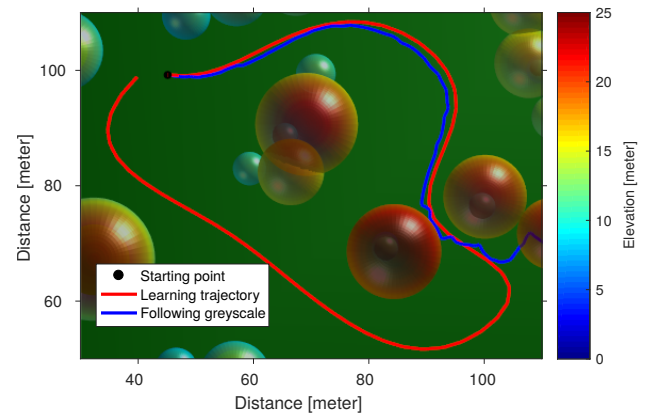
(b) Route following without noise ($\sigma = 0^\circ$, $r = 0$ m)(c) Failure for sky-segmented images ($\sigma = 10^\circ$, $r = 0$ m)(d) Failure for greyscale images (noise $\sigma = 10^\circ$, $r = 0$ m)(e) Failure for greyscale images ($\sigma = 10^\circ$, $r = 0$ m)

Figure 6. Route navigation with snapshots solely collected on the route trajectory ($r = 0$ m). (a) Rendered views (from different perspectives) of the 3D model used for simulation with the 200 m long route trajectory highlighted by light red spheres (distance between center of spheres is 1 m). (b) Without noise, the route, which is identically to the learning trajectory (red curve) here, can be successfully repeated using either greyscale images (blue curve) or sky-segmented images (green curve). (c,d,e) Examples of route following with noisy vectors ($\sigma = 10^\circ$). For each sub-figure different randomly generated error vectors were added during route learning. (c) In this example, route following with sky-segmented images fails due to significant deviations at a point where the route passes between two nearby trees. As sky-segmented images contain less texture information than greyscale images mismatches and therefore failures of route following are more likely. (d) Failure example of route following with greyscale images. Significant drift occurs after about 175 m caused by a large randomly generated error vector that finally leads to a mismatch. (e) Example of an early failure of route following with greyscale images due to mismatches caused by small deviations from route in combination with nearby trees. Note that for better visibility the simulated environment is illustrated in a schematic way in (b)-(e) and in the following figures, with the ground depicted in green and the objects, mainly trees, approximated by coloured spheres of different sizes whereby colour encodes elevation, i.e. the height above ground.

in Figs.7 a,b. By inspecting the sky-segmented images in figure 7 e, it is apparent that the skyline tends to flatten due to the high altitude of the agent (B) or to disappear since the agent moves directly between two trees (C). In both cases the skyline provides little information to align the agent with the correct snapshots and, in the worst case, the

current view may be matched with a completely unrelated snapshot. In contrast, the greyscale images still provide more visual information such that the agent is able to follow the route. Accordingly, as shown in figure 7 (e), the average distance to the route increases from 0.37 m to 0.50 m and from 0.32 m to 0.48 m for the route “low” and route “high”,

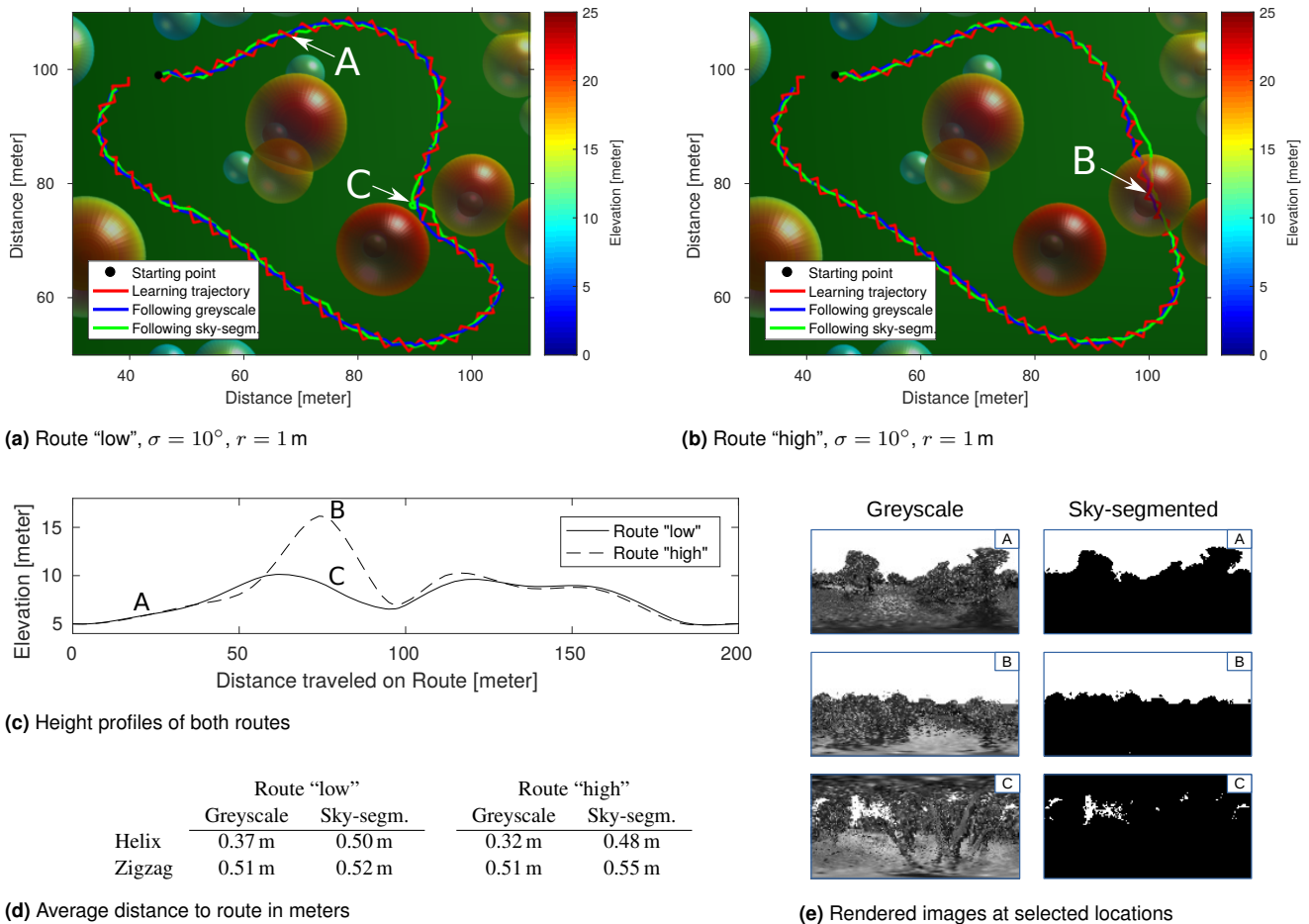


Figure 7. Examples and results of the proposed route navigation method. (a) The helix-like learning trajectory (red curve, helix radius $r = 1$ m) is shown for the same route as in Fig. 6. Since snapshots are available on all sides, lateral deviations from the route can be compensated. (b) Additional route variant (route "high") that in contrast to the original route passes over a tree (close to the position labeled 'B'). The height profile of both variants is shown in (c). (d) Average distance to route over 20 trials. All runs were successful, i.e. agent ended up within 2 m of last snapshot position on route. Deviations from both routes were larger for sky-segmented images compared to greyscale images and for the curved zigzag trajectory compared to the helix. (e) Examples of greyscale and sky-segmented images at selected positions, see corresponding labels 'A', 'B', 'C' in (a), (b) and (c).

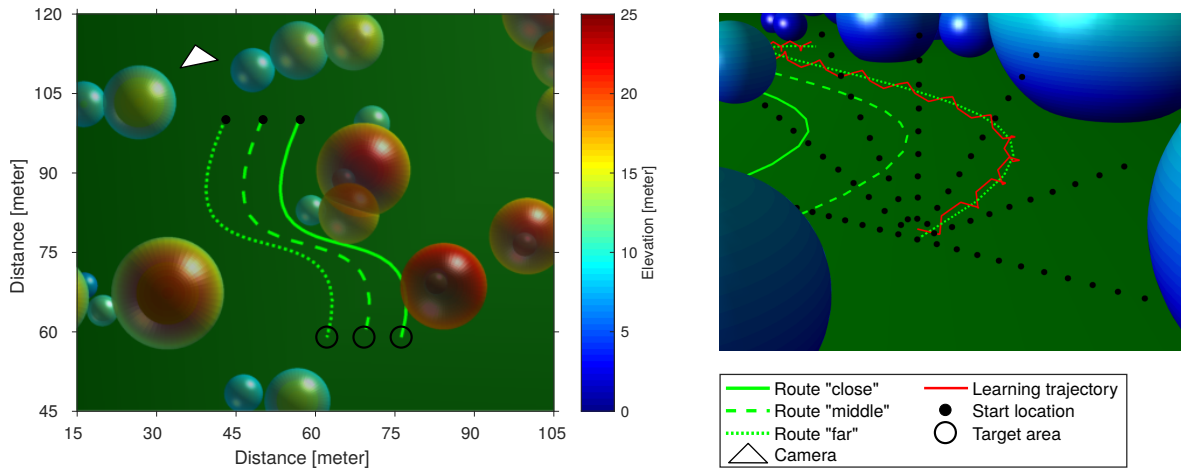
respectively. Moreover, it can be seen that using the curved zigzag trajectory to capture snapshots also increases the average distance to the route slightly. However, we think that the curved zigzag trajectory provides a very good trade-off between route following accuracy and complexity of the agent's movements during route learning.

5.2 Systematic Analysis

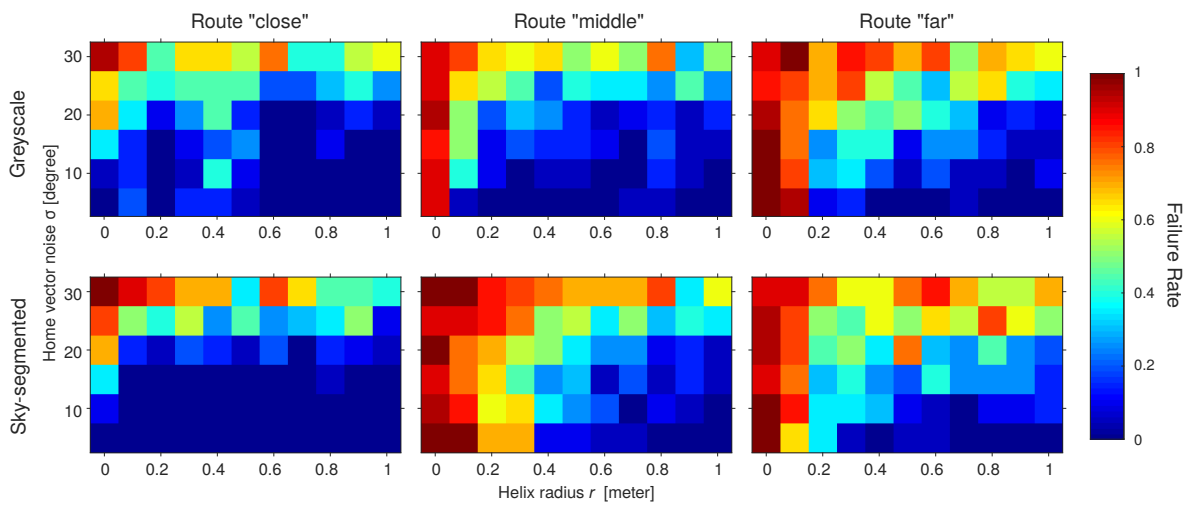
In order to study the impact of noise, varying helix radii, and translational offset from the route as well as the effect of nearby objects in the scene we performed systematic experiments.

Three routes of about 50 m length called "close", "middle", and "far" were generated, see figure 8 a. These routes differ by their start locations, which are shifted horizontally by 2 or 4 meters (routes "middle" and "far" with respect to route "close") such that the overall distance to the central group of trees increases. The elevation throughout the route is between 2 m to 4.5 m. In the first experiment of the systematic analysis, we study, by varying the helix radius r , how far the agent has to depart from the route to gather sufficient information from translational movement. To examine the influence of noise, noise was added to

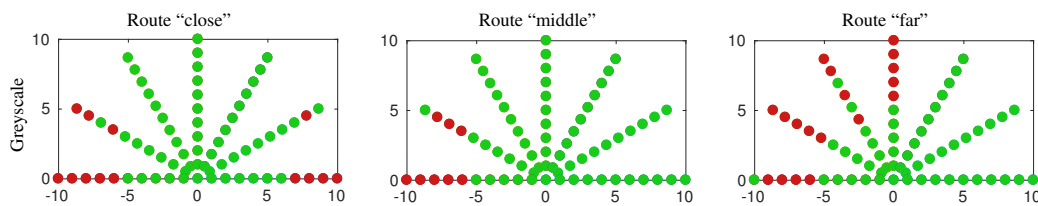
home vectors in dependence on standard deviation σ that was systematically increased. For each combination of $r \in \{0, 0.2, \dots, 1\}$ and $\sigma \in \{5^\circ, 10^\circ, \dots, 30^\circ\}$ 20 runs were simulated and the average failure rate determined. A route following run was counted as successful if the simulated agent arrived in a proximity of 2 m around the last location on the route (target area). The results are shown in figure 8 b for both greyscale and sky-segmented images. The plots show the fraction of successful route following attempts for given combinations of the parameters r and σ . For all shown results there is a tendency that the performance increases with radius r of the helix and decreases with standard deviation σ . This can be expected, since snapshots captured using a helix trajectory with a larger radius contain more information for detecting and compensating lateral displacement, while for increased home vector noise the agent is more likely to leave the route. The results confirm our previous observations (see section 5.1) that it is not sufficient to capture snapshots along the route but to use, for example, the helix trajectory instead. Interestingly, using sky-segmented images seems to work better than greyscale images for smaller helix radii on route "close" while the opposite is true for route "middle". A possible explanation is that the skyline along route "close" is



(a) Overview of the routes “far” (dotted), “middle” (dashed) and “close” (solid line)



(b) Effect of helix radius r and home vector noise σ



(c) Effect of translational offset ($r = 1$ m, $\sigma = 0^\circ$)

Figure 8. Setup and results of the systematic study. (a) Illustration of the three routes that are (in x -direction) shifted versions of each other. The white triangle (“camera”) in the left figure indicates the position from which the scene is viewed in the right figure that visualizes the different starting positions (for route “far” in this example). (b) Failure rate (colour coded) for each route in dependence on helix radius r and standard deviation of noise σ estimated over 20 trials. Results for both greyscale (top row) and sky-segmented images (bottom row) are presented. (c) Effect of initial offset on route following performance. Green and red dots indicate whether or not the target zone (sphere with 2 m radius around final position of route) could be reached. Helix radius r was set to 1 m. For each offset and route, only a single run was simulated as no noise was applied.

unique enough and the number of very close features that can lead to mismatches as their position in the image is extremely sensitive to translations is reduced in sky-segmented images compared to greyscale images. For route “middle” larger displacements are necessary for sky-segmented images as the treetops defining the skyline are further away compared to

route “close” while greyscale images can exploit additional features below the skyline.

In the second experiment of the systematic analysis, we examined the magnitude of lateral displacement from the route that can be tolerated by our approach. For this purpose, as shown in the right side of figure 8 a, the start location of the agent is offset orthogonally to the initial route in 1 m

steps up to a maximum of 10 m in 7 different directions. For this experiment the radius is set to $r = 1$ and noise is not applied. Note that the tested offsets (up to 10 m) are large in comparison to the helix radius. As can be seen in figure 8 c, the overall results indicate that – at least for the tested routes and parameters – offsets of up to 5 m can be compensated in almost all cases. The best results were achieved on the route “middle”; they become worse if surrounding objects are either closer (route “close”) or further (route “far”) away from the route. Interestingly, for the routes “close” and “middle” most failures appear for horizontal offset of the agent, while for the route “far” most failures appear for vertical offset. This could be due to the overall larger distance to objects on the route “far”, reducing the number of salient features at the skyline.

In summary, the systematic study supports the benefit of capturing snapshots around a route. In agreement with the results of the previous section, failure rate was significantly lower for larger radii as deviations from the route can be compensated, reducing the chance of mismatches. If mismatches occurred they were mainly caused by insufficient visual information or by nearby trees where textural information changes quickly with position.

6 Discussion

In this study we generalized the multi-snapshot model for navigation in three dimensions. Based on this generalization, we presented a novel approach for route following using panoramic images and verified its functionality in a simulated environment. The central idea of our approach is to collect snapshots not just on the route but with lateral displacements, which does not only allow to compensate of drift during route following very effectively, as we showed in section 5. Our approach only requires comparisons of images that may have different orientations, which we compute efficiently in the basis of spherical harmonics. As a consequence, our proposed method has only marginal requirements regarding computational power and memory. Motivated by the comparatively low resolution of insect eyes, see e.g. (Land and Nilsson 2012), we only use the first $l = 15$ frequencies to represent panoramic images in frequency domain by a vector with $l^2 = 225$ entries (section 4.2). By storing these vectors as floats, each snapshot only requires 900 B memory. Therefore the complete route of around 200 m as used in our examples, where a snapshot is stored after each meter, requires less than 200 kB.

Illumination-robust representations We also tested sky-segmented images as they have been suggested as a robust visual representation and could be realized in insects by means UV or UV/color contrast (Möller 2002; Stone et al. 2014; Differt and Möller 2015). While the performance of route-following was usually higher with greyscale images one has to consider that our experiments were done in a simulated environment and results might be different in case of varying lighting conditions, which we currently cannot render in a satisfactory way. Nevertheless, our results indicate that sky-segmented images can be used as an alternative to greyscale images since they are – at least theoretically – completely invariant against illumination

changes. The reduced amount of information stored in sky-segmented images increases the probability of mismatches. Therefore sky-segmented images are likely to benefit from using a higher number of frequencies l to represent especially the skyline in more detail.

Sequence-based matching In contrast to most previous insect-inspired models of route navigation that represent the complete route as an unordered set of snapshots, e.g. (Baddeley et al. 2012; Kodzhabashev and Mangan 2015; Gaffin and Brayfield 2016) or in a “holistic memory”, i.e. a neural network or classifier that, after being trained on a set of snapshots, assesses the familiarity of views (Baddeley et al. 2011, 2012; Wystrach et al. 2013; Ardin et al. 2016; J. Müller et al. 2018), we use the sequence information to confine the search range for the best-matching snapshot (± 20 snapshots around the previously best-matching snapshot). Note that in case route following should also be possible starting at any positions near the route and not just at the beginning, we could extend our model by an initial global search for the best match. Furthermore, the search range could be continuously adapted depending on match confidence. While the limited search range already reduces computation time and the chance of mismatches, further improvements might be possible by enforcing a stricter sequence-based matching as, for instance, in (Milford and Wyeth 2012; Pepperell et al. 2014; Grix et al. 2018). This could be helpful for sky-segmented images that usually provide less textural information than greyscale images but also in situations with ambiguous visual input or significant change in environment appearance between route learning and following. By matching sequences, Milford (2013) could show that robust place recognition can be achieved even with very low-resolution images. Even though a number of experiments suggest that insects do not rely on sequence information (Collett et al. 1992; Kohler and Wehner 2005; Mangan and Webb 2012), more experiments are necessary to clarify this matter as a sequence-based method would clearly be useful in ambiguous locations (two visually indistinguishable locations are associated with different movement directions) or in environments with strong changes in appearance.

Benefits and limits of spherical harmonics Representing panoramic images that cover the full viewing sphere (4π solid angle) in spherical harmonics is not only memory efficient but also ideal for orientation and similarity estimation. Based on these properties, we proposed a general and efficient solution for route navigation where relative camera orientation can be arbitrary, i.e. very large rotation differences in all DOFs (yaw, pitch, roll) between route snapshots and views during route repetition can occur. For instance, it would work even for agents that look in opposite directions during route learning and following or continuously rotate around their body center. Note that, as described in section 4, the orientation search range of the visual 3D compass was limited in our implementation. However this was done only for reducing the chance of mismatches and enhancing efficiency further since smooth trajectories were used for route learning.

For route reversal, i.e. for following the route from its end point to the start, one just has to reverse the sequence, flip the

vector components parallel to the route, i.e. replace \vec{v}_i by $-\vec{v}_i$ in Eq. (9), and, in case search range of the visual compass is limited, center the yaw angle search range at 180° .

However, for reduced field of view, spherical harmonics are less suited. While masking out the missing zones is possible (Differt 2017a), it makes the approach less efficient and a different approach might be favourable. In case pitch and roll angles are stabilized as has been found in flying insects (Boeddeker and Hemmi 2010; Viollet and Zeil 2013; Mureli et al. 2017), orientation estimation is basically reduced to one-dimensional search (yaw angle), which can be done efficiently directly on low-resolution images. If, in addition, the field of view is symmetric around the yaw-axis, Zernike polynomials might be an interesting option (Stone et al. 2018).^{*} The search space can be further reduced, if, as in flying insects (Zeil 1993; Boeddeker et al. 2010; Philippides et al. 2013), head or camera orientation can be decoupled at least to some extent from flight direction, allowing to align snapshots and views along the route during learning and following, as illustrated in last row of Fig. 2 c.

Relevance for flying insects Our approach is based on the multi-snapshot model which was motivated by observations in walking insects (Graham et al. 2010; Narendra et al. 2013; Dewar et al. 2014). As we extended it to 3D the question arises about its relevance for modelling learning and return flights. The visual homing model of Stürzl et al. (2016) for ground-nesting wasps assumes that nest tagged views, i.e. views with the directional information “nest left” or “nest right”, are captured during learning flights guiding the wasp when returning to the nest by the directions linked to the best matching snapshots. This is an example where home vectors are associated to snapshots that are not aligned with the nest direction and where snapshots are not just captured at ground level but at different heights. However, movements in the vertical dimension are not directly controlled by snapshot similarity and vectors due to the fact that the nest is at the ground and, therefore, no snapshots below the goal (with vectors pointing upwards) are captured. Note that the situation will be different for learning flights in the vicinity of a feeder placed on a vertical wall or at the edge of a table as has been observed in bees (Lehrer 1993).

We then, based on the 3D multi-snapshot model, developed a route navigation model with a helix trajectory for route learning. We showed in section 5.1 that a curved zigzag trajectory, which seems to be more suitable model for flying insects, can be used without significant loss of performance. In case the distance to ground is approximately known or can be controlled by other means only snapshots to the left and the right of the route have to be collected and a simple sideways zigzag movement would be sufficient. While zigzag movements have been observed in a variety of flying insects, for instance in wasps (Zeil 1993; Voss and Zeil 1998) or bees (Philippides et al. 2013), and recently even a “zigzag module” has been identified in the lateral accessory lobe (Namiki and Kanzaki 2016), it is unclear whether sideways movements are performed on routes over larger distances. Although harmonic radar is a powerful tool for recording flights over hundreds of meters, see for example (Capaldi et al. 2000), it unfortunately does not reveal such details. Therefore, we can currently only hypothesize how route learning as described by our

model could be realized in flying insects. While spherical harmonics allow to efficiently compare images that have large orientation differences, insects might recognize a snapshot only if it is aligned with the current view, possibly within a few degrees. However, flight direction is up to a certain degree independent of head orientation in most flying insects (Zeil 1993; Boeddeker et al. 2010; Philippides et al. 2013). Therefore, as mentioned before, they could efficiently collect snapshots on a curved zigzag path by keeping the head aligned with the route. Nevertheless, for being able to reverse a route and return to the nest after a longer exploratory flight, this is not sufficient and, from time to time, the insect would have to turn on its outbound trip and acquire snapshots oriented in the opposite direction.

How could insects estimate the vectors associated with snapshots? For learning walks of ants, it has been suggested that the direction to the goal is provided by path integration (Müller and Wehner 2010). Ground-nesting wasps might visual track their nests during learning flights (Samet et al. 2014; Schulte et al. 2019). For 3D route learning, directions could be obtained by means of local path integration but also, in particular if the helix or curved zigzag trajectories are quite stereotyped, the vectors could be pre-set or learned and simply recalled during flight. Our simulation results show that these vectors do not have to be highly accurate if an appropriate radius is chosen.

Adapting the learning trajectory In the current implementation snapshots were taken at constant intervals. While it is obvious that this could be done more efficiently by recording only snapshots necessary for robust route navigation, it is less clear how this objective could be implemented. Monitoring the similarity to the most recent snapshot (Franz et al. 1998a; Smith et al. 2007; Zeil et al. 2010), possibly in relation to the similarity of other snapshots, might be a good starting point. The distance between neighboring snapshots will be related to the extent of their catchment volumes (Murray and Zeil 2017) in the respective directions but will also depend on the “environmental noise”. In addition, the image resolution could be continuously adapted, which in our approach can be done by changing the number of frequencies l .

For recording snapshots with lateral offset from the route in an efficient way we proposed to use a helix or curved zigzag trajectory. In section 5.2 the effect of the helix radius was studied by testing different values that, however, were fixed during each run. Actively controlling the radius (and possibly the “wavelength” λ , as well) would allow to smoothly adapt the desired trajectory to the environment. For instance, the radius could be adjusted according to the distances to surrounding objects but also in dependence on the maximum deviation from the route that is to be expected or can be tolerated. Furthermore, capturing snapshots with offset from the route might be necessary only at certain critical locations, which can be achieved by setting the

^{*}Note that Zernike polynomials, similar to image representations using (non-spherical) Fourier coefficients (Pajdla and Hlaváč 1999; Menegatti et al. 2004; Stürzl and Mallot 2006) are only recommended in case of negligible variation in pitch and roll angles (i.e. predominantly yaw rotations). For general rotations that include significant pitch and roll angles, spherical harmonics are preferable, at least for fully spherical images.

radius to a small or even zero value for most of the time, thereby keeping the learning trajectory close to the route, and increasing the radius to its maximum size just before such locations (as well as decreasing it afterwards). Interestingly, [Baddeley et al. \(2012\)](#) included learning walks around the nest in their route following model in order to avoid simulated ants overshooting the goal. Similar to our approach, these additional views also allow to reduce deviations from the route, although only near the goal.

The selection of snapshots along a route in space and time is an exciting topic that has to be addressed in future work. Most likely a complete and general set of rules for controlling these parameters in a way that guarantees robust and efficient route navigation in all environments will be impossible to find. Thus, assessing the quality of the acquired route representation already during learning while information is available by other means, e.g. from path integration, might be essential.

It would also be interesting to investigate whether trajectories similar to the one used for learning could also be employed for following a route, in particular those defined by a “straight” learning trajectory as in the first experiment of section 5.1. Based on the change in similarity of the most similar snapshots with the views experienced along the path, one could continuously adjust the center of the trajectory. However, as this would have to be done in every route repetition, using these trajectories during learning seems to be more efficient. Another option is to combine both strategies using, for instance, a mixture of sporadic larger displacements during learning by controlling radius r as discussed before and curved zigzag movements of smaller amplitudes during route following. The latter will in addition enhance obstacle detection by means of optic flow.

Ultimately, the proposed approach and possible extensions for route navigation in 3D have to be tested in diverse environments. For this purpose, an implementation on a flying system such as a multicopter ([M.G. Müller et al. 2018](#)), is planned.

Declaration of conflicting interests

The authors declare that there is no conflict of interests.

Funding

This work was supported by the German Science Foundation (DFG), grant numbers MO 1037/10-1 and STU 413/2-1, and by the Australian Research Council, grant number DP150102699.

References

- Ardin P, Peng F, Mangan M, Lagogiannis K and Webb B (2016) Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLOS Computational Biology* 12(2): 1–22.
- Baddeley B, Graham P, Husbands P and Philippides A (2012) A model of ant route navigation driven by scene familiarity. *PLOS Computational Biology* 8(1): 1–16.
- Baddeley B, Graham P, Philippides A and Husbands P (2011) Holistic visual encoding of ant-like routes: Navigation without waypoints. *Adaptive Behavior* 19: 3–15.
- Bishop CM (2006) *Pattern Recognition and Machine Learning*. Springer.
- Boeddeker N, Dittmar L, Stürzl W and Egelhaaf M (2010) The fine structure of honeybee head and body yaw movements in a homing task. *Proceedings of the Royal Society B: Biological Sciences* 277(1689): 1899–1906.
- Boeddeker N and Hemmi JM (2010) Visual gaze control during peering flight manoeuvres in honeybees. *Proceedings of the Royal Society B: Biological Sciences* 277(1685): 1209–1217.
- Capaldi E, D Smith A, Osborne J, Fahrbach S, M Farris S, Reynolds D, S Edwards A, Martin A, Robinson G, Poppy G and R Riley J (2000) Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature* 403: 537–540.
- Cartwright B and Collett T (1987) Landmark maps for honeybees. *Biological Cybernetics* 57: 85–93.
- Cartwright BA and Collett TS (1983) Landmark learning in bees: Experiments and models. *Journal of Comparative Physiology* 151(1): 521–543.
- Collett TS, Dillmann E, Giger A and Wehner R (1992) Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A* 170: 435–442.
- Dewar AD, Philippides A and Graham P (2014) What is the relationship between visual environment and the form of ant learning-walks? An in silico investigation of insect navigation. *Adaptive Behavior* 22(3): 163–179.
- Differt D (2017a) *Holistic Methods for Visual Navigation of Mobile Robots in Outdoor Environments*. PhD Thesis, Bielefeld University, Germany.
- Differt D (2017b) Real-time rotational image registration. In: *International Conference on Advanced Robotics (ICAR)*. pp. 1–6.
- Differt D and Möller R (2015) Insect models of illumination-invariant skyline extraction from UV and green channels. *Journal of Theoretical Biology* 380(7): 444–462.
- Differt D and Möller R (2016) Spectral skyline separation: Extended landmark databases and panoramic imaging. *Sensors* 16(10): 1–23.
- Fleer D and Möller R (2017) Comparing holistic and feature-based visual methods for estimating the relative pose of mobile robots. *Robotics and Autonomous Systems* 89(1): 51–74.
- Franz M, Schölkopf B, Mallot H and Bühlhoff H (1998a) Learning view graphs for robot navigation. *Autonomous Robots* 5: 111–125.
- Franz MO, Schölkopf B, Mallot HA and Bühlhoff HH (1998b) Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics* 79(1): 191–202.
- Gaffin DD and Brayfield BP (2016) Autonomous visual navigation of an indoor environment using a parsimonious, insect inspired familiarity algorithm. *PLOS ONE* 11(4): 1–25.
- Gaffin DD, Dewar A, Graham P and Philippides A (2015) Insect-inspired navigation algorithm for an aerial agent using satellite imagery. *PLOS ONE* 10(4): 1–14.
- Graham P (2010) Insect navigation. *Encyclopedia of Animal Behavior* 2(1): 167–175.
- Graham P, Philippides A and Baddeley B (2010) Animal cognition: Multi-modal interactions in ant learning. *Current Biology* 20(15): R639–R640.
- Grixa I, Schulz P, Stürzl W and Triebel R (2018) Appearance-based along-route localization for planetary missions. In:

- International Conference on Intelligent Robots and Systems (IROS)*. pp. 6327–6334.
- Judd SPD and Collett TS (1998) Multiple stored views and landmark guidance in ants. *Nature* 392(1): 710–714.
- Kodzhabashev A and Mangan M (2015) Route following without scanning. In: *Biomimetic and Biohybrid Systems*. pp. 199–210.
- Kohler M and 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.
- Land MF and Nilsson DE (2012) *Animal Eyes*. 2nd edition. Oxford University Press.
- Lehrer M (1993) Why do bees turn back and look? *Journal of Comparative Physiology A* 172(5): 549–563.
- Mangan M and Webb B (2012) Spontaneous formation of multiple routes in individual desert ants (*Cataglyphis velox*). *Behavioral Ecology* 23(5): 944–954.
- Menegatti E, Maeda T and Ishiguro H (2004) Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems* 47(4): 251–267.
- Milford M (2013) Vision-based place recognition: How low can you go? *International Journal of Robotics Research* 32(7): 766–789.
- Milford M and Wyeth GF (2012) SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In: *International Conference on Robotics and Automation (ICRA)*. pp. 1643–1649.
- Möller R (2002) Insects could exploit UV-green contrast for landmark navigation. *Journal of Theoretical Biology* 214(4): 619–631.
- Möller R (2009) Local visual homing by warping of two-dimensional images. *Robotics and Autonomous Systems* 57(1): 87–101.
- Möller R, Horst M and Fleer D (2014) Illumination tolerance for visual navigation with the holistic min-warping method. *Robotics* 3(1): 22–67.
- Möller R, Krzykowski M and Gerstmayr L (2010) Three 2D-warping schemes for visual robot navigation. *Autonomous Robots* 29(3): 253–291.
- Müller J, Nawrot M, Menzel R and Landgraf T (2018) A neural network model for familiarity and context learning during honeybee foraging flights. *Biological Cybernetics* 112(1-2): 113–126.
- Müller M and Wehner R (2010) Path integration provides a scaffold for landmark learning in desert ants. *Current Biology* 20(15): 1368–1371.
- Müller MG, Steidle F, Schuster MJ, Lutz P, Maier M, Stoneman S, Tomić T and Stürzl W (2018) Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision. In: *International Conference on Intelligent Robots and Systems (IROS)*. pp. 3701–3708.
- Mureli S, Thanigaivelan I, Schaffer ML and Fox JL (2017) Cross-modal influence of mechanosensory input on gaze responses to visual motion in drosophila. *Journal of Experimental Biology* 220(12): 2218–2227.
- Murray T and Zeil J (2017) Quantifying navigational information: The catchment volumes of panoramic snapshots in outdoor scenes. *PLOS ONE* 12(10): 1–24.
- Namiki S and Kanzaki R (2016) The neurobiological basis of orientation in insects: insights from the silkworm mating dance. *Current Opinion in Insect Science* 15: 16–26.
- Narendra A, Gourmaud S and Zeil J (2013) Mapping the navigational knowledge of individually foraging ants *Myrmecia croslandi*. *Proceedings of the Royal Society B: Biological Sciences* 280(1765): 20130683.
- Nicholson D, Judd S, Cartwright B and Collett T (1999) Learning walks and landmark guidance in wood ants (*Formica rufa*). *Journal of Experimental Biology* 202(13): 1831–1838.
- Pajdla T and Hlaváč V (1999) Zero phase representation of panoramic images for image based localization. In: *Computer Analysis of Images and Patterns (CAIP)*. pp. 550–557.
- Pepperell E, Corke P and Milford M (2014) All-environment visual place recognition with SMART. In: *International Conference on Robotics and Automation (ICRA)*. pp. 1612–1618.
- Philippides A, de Ibarra NH, Riabinina O and Collett TS (2013) Bumblebee calligraphy: the design and control of flight motifs in the learning and return flights of *Bombus terrestris*. *Journal of Experimental Biology* 216(6): 1093–1104.
- Samet N, Zeil J, Mair E, Boeddeker N and Stürzl W (2014) Ground-nesting insects could use visual tracking for monitoring nest position during learning flights. In: *From Animals to Animats 13*. pp. 108–120.
- Schulte P, Zeil J and Stürzl W (2019) An insect-inspired model for acquiring views for homing. *Biological Cybernetics* 113: 439–451.
- Smith L, Philippides A, Graham P, Baddeley B and Husbands P (2007) Linked local navigation for visual route guidance. *Adaptive Behavior* 15(3): 257–271.
- Srinivasan MV (2011) Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. *Physiological Reviews* 91(2): 413–460.
- Stone T, Mangan M, Ardin P and Webb B (2014) Sky segmentation with ultraviolet images can be used for navigation. In: *Robotics: Science and Systems (RSS)*.
- Stone T, Mangan M, Wystrach A and Webb B (2018) Rotation invariant visual processing for spatial memory in insects. *Interface Focus* 8: 20180010.
- Stürzl W, Grixa I, Mair E, Narendra A and Zeil J (2015) Three-dimensional models of natural environments and the mapping of navigational information. *Journal of Comparative Physiology A* 201(6): 563–584.
- Stürzl W and Mallot HA (2006) Efficient visual homing based on Fourier transformed panoramic images. *Robotics and Autonomous Systems* 54(4): 300–313.
- Stürzl W, Zeil J, Boeddeker N and Hemmi JM (2016) How wasps acquire and use views for homing. *Current Biology* 26(4): 470–482.
- Vardy A and Möller R (2005) Biologically plausible visual homing methods based on optical flow techniques. *Connection Science* 17: 47–89.
- Viollet S and Zeil J (2013) Feed-forward and visual feedback control of head roll orientation in wasps (polistes humilis, vespidae, hymenoptera). *Journal of Experimental Biology* 216(7): 1280–1291.
- Voss R and Zeil J (1998) Active vision in insects: an analysis of object-directed zig-zag flights in wasps (*Odynerus spinipes*,

- Eumenidae). *Journal of Comparative Physiology A* 182: 377–387.
- Wolf H (2011) Review: Odometry and insect navigation. *Journal of Experimental Biology* 214(4): 1629–1641.
- Wystrach A and Graham P (2012) What can we learn from studies of insect navigation? *Animal Behaviour* 84(1): 13–20.
- Wystrach A, Mangan M, Philippides A and Graham P (2013) Snapshots in ants? new interpretations of paradigmatic experiments. *Journal of Experimental Biology* 216(10): 1766–1770.
- Wystrach A, Philippides A, Aurejac A, Cheng K and Graham P (2014) Visual scanning behaviours and their role in the navigation of the Australian desert ant *Melophorus bagoti*. *Journal of Comparative Physiology A* 200(7): 615–626.
- Zeil J (1993) Orientation flights of solitary wasps (*Cerceris*; Sphecidae; Hymenoptera): I. Description of flight. *Journal of Comparative Physiology A* 172: 189–205.
- Zeil J (2012) Visual homing: An insect perspective. *Current Opinion in Neurobiology* 22(2): 285–293.
- Zeil J, Boeddeker N and Stürzl W (2010) Visual homing in insects and robots. In: Floreano D, Zufferey JC, Srinivasan MV and Ellington C (eds.) *Flying Insects and Robots*. Springer, pp. 87–100.
- Zeil J and Fleischmann PN (2019) The learning walks of ants (Hymenoptera: Formicidae). *Myrmecological News* 29: 93–110.
- Zeil J, Hofmann MI and Chahl JS (2003) Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A* 20(3): 450–469.
- Zeil J, Narendra A and Stürzl W (2014) Looking and homing: How displaced ants decide where to go. *Philosophical Transactions of the Royal Society B: Biological Sciences* 369: 20130034.