*Connection Science*
Vol. 17, Nos. 1–2, March–June 2005, 47–89

# Biologically plausible visual homing methods based on optical flow techniques

ANDREW VARDY*† and RALF MÖLLER‡

†School of Computer Science, Carleton University, Ottawa, Canada
‡Computer Engineering Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany

Insects are able to return to important places in their environment by storing an image of the surroundings while at the goal, and later computing a home direction from a matching between this 'snapshot' image and the currently perceived image. Very similar ideas are pursued for the visual navigation of mobile robots. A wide range of different solutions for the matching between the two images have been suggested. This paper explores the application of optical flow techniques for visual homing. The performance of five different flow techniques and a reference method is analysed based on image collections from three different indoor environments. We show that block matching, two simple variants of block matching and two even simpler differential techniques produce robust homing behaviour, despite the simplicity of the matched features. Our analysis reveals that visual homing can succeed even in the presence of many incorrect feature correspondences, and that low-frequency features are sufficient for homing. In particular, the successful application of differential methods opens new vistas on the visual homing problem, both as plausible and parsimonious models of visual insect navigation, and as a starting point for novel robot navigation methods.

*Keywords*: Robotics; Navigation; Optical flow; Biological plausibility

## 1. Introduction

Biologists have studied the amazing ability of insects such as bees, ants and wasps to return to important places in their environment after displacement. It is thought that visual homing may be an important tool in the navigational repertoire of these insects (Collett 1996, Collett and Collett 2002). Visual homing is the act of returning to a goal position by comparing the image currently viewed with a 'snapshot' image taken when at the goal. Various models have appeared in the biological literature that provide an algorithmic-level description of how insects might use visual homing strategies to return to a goal position (Anderson 1977, Cartwright and Collett 1983, 1987).

Robotics has begun to adopt general aspects of animal homing strategies, specifically the idea that navigation is not necessarily identical with the problem of determining position and orientation with respect to a map, but can be accomplished more directly by using local homing strategies to reach some location. Local homing strategies are building blocks of

*Corresponding author. Email: avardy@engr.mun.ca
Presently assistant Professor of Electrical and Computer Engineering, Memorial University of Newfoundland, St. Johns, Canada.

*Connection Science*
ISSN 0954-0091 print/ISSN 1360-0494 online © 2005 Taylor & Francis Group Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/09540090500140958

topological approaches to navigation. Topological approaches represent the world as a graph, with nodes corresponding to locations, each characterized by a sensory signature, and edges establishing relations between locations such as reachability or proximity (Kuipers and Byun 1988, Kortenkamp 1993, Franz *et al.* 1998a, Franz and Mallot 2000).

In this context, engineers have designed a number of visual homing algorithms for mobile robots that were inspired by biological models. One biological model that has received a great deal of attention from robotics is the *snapshot model* (Cartwright and Collett 1983, 1987). Algorithms based on the snapshot model propose various strategies for finding features within images and establishing correspondence between them in order to determine a home direction (some of these algorithms are reviewed later in this paper). Thus, visual homing methods are closely related to the problem of measuring ego-motion from optical flow (Gluckman and Nayar 1998).

Interestingly, classical optical flow techniques have received relatively little attention as a means of corresponding features for visual homing. One such technique is *block matching* (Jain and Jain 1981). Block matching takes a block of pixels from one image and searches for the best matching block in another image within a fixed search radius. The degree of match between blocks is usually judged by the sum of squared differences (SSD) or some other local correlation measure. Variants of block matching are widely used in computer vision to determine the motion between frames of a video sequence so as to allow more efficient encoding. In this paper, we present an implementation of block matching for visual homing. Our experiments reveal this method to be highly successful on both offline images and for online control of a mobile robot.

An analysis of block matching's success leads us to propose two simpler variants that are also found to be successful for homing. These two methods have reduced computational complexity from block matching, but still require an extensive local search process for every correspondence. We therefore continued by investigating differential techniques for optic flow. First- and second-order differential methods are proposed that do not require any correspondence search whatsoever. We feel that, of all the methods presented here, the differential methods are simple enough and require the right style of processing (*i.e.* retinotopic) to be plausible as models for insect visual homing.

In addition to proposing these techniques for finding correspondences between images, we present two novel methods for transforming such correspondences into home vectors. The first method does not require any approximations. Indeed, the use of both dimensions of the input image allows this method to avoid making any assumptions about the distribution of landmarks in the environment (such as those made by Franz *et al.* (1998b) to handle one-dimensional images). The second method is utilized for the differential methods that estimate the direction of correspondence, but not the length of the correspondence vector. Consequently, this second method is based on some reasonable approximations.

We employed a number of strategies to test our homing methods. The block matching method was tested most thoroughly, using both offline and online experiments on images from a robot-mounted panoramic imaging system. Other methods were tested in offline experiments – whose results appear to agree with online experiments. Images for offline testing were obtained in an unmodified office environment under various lighting conditions and in the presence of obstacles. Images were also collected in two other qualitatively different indoor environments to test the transferability of our results to other environments. Comparisons are made with Franz *et al.*'s (1998b) *warping method*, one of the most highly cited methods for visual homing that various other authors have also used for comparison (Weber *et al.* 1999, Stürzl and Mallot 2002, Vardy and Oppacher 2003, 2004, 2005).

In section 2, we present a review of visual homing strategies for mobile robots. We then introduce the image database and experimental methodology used throughout this paper in section 3. Our block matching method is described in section 4, along with our technique for mapping correspondence vectors to home vectors. Also, results are presented on this method and an analysis is carried out to explain its good performance. In section 5, we present two simpler variants of block matching and show results on these. In section 6, the differential methods are presented and tested. A discussion of the significance of our results follows in section 7. Finally, concluding remarks are made in section 8. Two appendices follow the main text. Appendix A presents the derivation of the differential methods. Appendix B provides additional detail on the exact vector mapping technique and presents the entirety of the approximate technique used by the differential methods.

## 2. Visual homing strategies for mobile robots

Approaches to visual homing of mobile robots have their roots in several fields: some arise from the plethora of methods in computer and robot vision, while others are either inspired by animal homing strategies, or are even specifically aimed at subjecting biological models to a strict real-world test (Webb 2001). Franz *et al.* (1998b) and Franz and Mallot (2000) use the terms *guidance* and *recognition-triggered response* to distinguish between two classes of visual homing mechanisms. Guidance approaches (also referred to as image matching) are local homing methods that determine the direction to a single goal by comparing a snapshot image taken at the goal location with the image perceived at the current location. The term recognition-triggered response describes a class of methods where *associations* between multiple images and either a home direction or a position is learned in an exploration phase; the associations are later used to recall the home direction or the position from the current image (Nelson 1989, Cassinis *et al.* 1996, Jogan and Leonardis 1999, Gaussier *et al.* 2000).

In the following, we shall focus on the class of guidance methods to which our models belong. Many of these approaches, including ours, relate to the snapshot model (Cartwright and Collett 1983, 1987) which originally was devised to explain the results of landmark navigation experiments on honeybees. In general, guidance strategies have to derive a home direction from the discrepancies between a single snapshot image and the current image. Both images are assumed to be static and, in most cases, panoramic snapshots without explicit depth information. Approaches extracting depth information from multiple cameras or special sensors (Stürzl and Mallot 2002) are less plausible as models of insect navigation, since insects are prevented from effectively using stereo vision due to the small distance between their eyes. We therefore exclude models relying on distance information (like the recurrent neural network model suggested by Cruse (2003)) from the discussion.

With few exceptions (to be discussed later), most guidance approaches select certain features in the snapshot image as *landmarks*, and try to establish correspondences between these landmarks and features extracted from the current view image. Correspondence approaches differ with respect to the strategy for selecting the landmarks. While some methods strive to extract maximally distinctive features (and thus can be said to *identify* landmarks) (Hong *et al.* 1992, Se *et al.* 2001, Lowe 2004), other approaches use less unique features such as dark or bright sectors (Cartwright and Collett 1983, 1987, Weber *et al.* 1999, Lambrinos *et al.* 2000), Harris corners (Vardy and Oppacher 2003, 2004), or coloured regions (Gourichon *et al.* 2002, Goedemé *et al.* 2004). Methods based on distinctive landmarks ideally succeed in establishing one-to-one correspondences by searching for each landmark feature in the current view; if the search covers the entire image, there is no need to align the views to the same external coordinate system (and therefore no need for some type of compass reference). In contrast, when using non-unique landmarks, the search will produce many-to-many mappings between landmark features in the snapshot and in the current view, which may have to be resolved by optimization processes such as dynamic programming (Gourichon *et al.* 2002) or other correspondence methods (Weber *et al.* 1999). Usually, the correspondence problem cannot be solved with sufficient reliability if the views are not aligned to the same global coordinate system beforehand in order to bring corresponding features into roughly the same bearing – at least approximate compass information is required.

The performance of correspondence methods, specifically of those using distinctive features, depends crucially on the invariance of the extracted features to scaling, distortion and changes in illumination. Computer vision approaches go to great lengths to extract invariant features, *e.g.* by using scale spaces and orientation histograms (Se *et al.* 2001, Lowe 2004), projective invariants derived from a number of coplanar feature points (Colios and Trahanias 2001), or invariant descriptors computed through the Fourier–Mellon transform (Rizzi *et al.* 2001). Moreover, landmark features should be robust under movements of the robot; selecting visual features encompassing spatially separate objects should specifically be avoided. Therefore, some robot methods (Lehrer and Bianco 2000) resort to a strategy observed in bees and wasps: when departing from a novel goal location, these insects perform stereotypic flight manœuvres called ‘turn-back-and-look’ behaviour – during these movement sequences, robust landmarks could be selected by a tracking process. Specifically for outdoor navigation, invariance against changes in illumination becomes an important property; colour opponency mechanisms have been suggested as a possible solution, from both the technical and the biological perspective (Möller 2002b, Todt and Torras 2004). However, a small number of guidance methods deviate from the concept of landmark extraction and matching. In this work, we use the warping method of Franz *et al.* (1998b) as a reference method since it has proven its value with respect to homing performance and robustness against parameter selection in a number of previous experiments (Weber *et al.* 1999, Möller 2002a, Stürzl and Mallot 2002, Vardy and Oppacher 2003, 2004, 2005). In the warping method, a one-dimensional panoramic image (here the current image) is warped according to three parameters that describe the direction, distance and rotation of the robot’s movement from the goal position to its current position. This image warp is based on the *equal distance assumption*, which means that all landmarks are located at roughly the same distance from the goal. Although this assumption will usually be violated, Franz *et al.* show that the error due to this assumption decreases with distance to the goal, under the additional assumption that the agent is in the open space around the goal. The warped image approximates the view that the robot would have perceived if it had moved as described by the three parameters. In other words, relative to the agent’s current position, these parameters specify a putative goal position. This putative goal is evaluated by comparing the warped image with the snapshot image. The home direction is obtained by an exhaustive minimum search through the parameter space with respect to a distance measure between the warped and snapshot images. Since the rotation of the robot is one of the parameters, the method can dispense with an external compass reference. We shall refer to our implementation of Franz *et al.*’s method as ‘Warping’.

A second guidance method that still relies on landmark extraction but does not require a matching process is the ‘average landmark vector (ALV) model’ developed by Lambrinos (1999) and Lambrinos *et al.* (2000). This method can be characterized as a parameter method, in contrast to the previously described methods, which are referred to as template methods. While template methods use an entire image as the ‘signature’ of a place, parameter methods derive a compact description from the image comprising only few parameters. Whether insects use templates or parameters or both is still an unresolved question (Möller 2001). The average landmark vector method characterizes each location by the two components of the average landmark vector, which is computed by averaging unit vectors pointing towards all landmarks, and the home direction is determined simply by subtracting two average landmark vectors (expressed with respect to the same coordinate system by using a compass). Interestingly, the average landmark vector model turned out to be a mathematical simplification of the snapshot model (Möller 2000). However, the weak point of this method is its high demands on the robustness of landmark detection. While the ALV model has worked well with artificial landmarks (Möller 2000, 2002c), it has so far failed to reach the performance of the warping method, despite attempts to make the landmark detection more robust (Möller *et al.* 2001). A neural version of the ALV model proposed by Hafner (2001) has shown good performance in some environments, but relies upon the environment exhibiting a stable visual centre-of-mass which does not appear to exist for at least two of the environments tested here (Vardy 2005).

A third model without image matching has recently appeared that is both surprisingly simple and effective. Zeil *et al.* (2003) showed that the home direction can be determined from the gradient of the root mean square (RMS) difference between the current image and the snapshot. This clearly requires a compass, but also the compass information can be derived from the minimum of a rotational RMS function. In experiments on the image databases used in this paper, this extremely parsimonious method turned out to perform surprisingly well; however, it did not exceed the performance of the warping method (Vardy 2005). One disadvantage of the RMS method is the necessity to approximate the gradient of the RMS function by sampling several points in space by exploratory movements of the robot. Also, this method can exhibit local minima in the RMS difference function which can impair performance.

The methods described in this work have their roots in optical flow techniques. While the optical flow method of *block matching* can be seen as an instance of the correspondence methods described above in general and of the snapshot model in particular, differential optical flow techniques have not, to the best of our knowledge, been applied to the visual homing problem before. Moreover, our methods do not require a pre-selection of landmark features, but rather treat each image region as a landmark.

## 3. Materials and methods

### 3.1 *Image database*

A database of images was collected in the robotics laboratory of the Computer Engineering Group of Bielefeld University. Images were collected by a camera mounted on a robot and pointed upwards at a hyperbolic mirror[†]. The hyperbolic mirror expanded the camera's field of view to allow the capture of panoramic images. Figure 1 shows the robot, and a close-up of the camera and mirror assembly.

The room was unmodified except to clear the floor and apply controlled additions to the original environment, as described below. The capture grid had dimensions 2.7 × 4.8 m, which covered nearly all of the floor's free space. Images were collected by positioning the robot along a line of 10 start positions spaced 30 cm apart. At each start position the robot was manually oriented with the long axis of the room by using two laser pointers mounted to the robot's chassis. After positioning the robot, the experimenter would leave the room and activate the robot's capturing cycle via wireless ethernet. At the start of the capturing cycle the robot would capture an image from the start position. It would then move ahead 30 cm, stop, capture another image and continue in this manner until it had reached the other side of the room.

The images were captured and stored at the camera's highest resolution (752 × 564). A hyperbolic mapping was then applied to produce panoramic images. These panoramic images

[†] The camera was an ImagingSource DFK 4303. The robot was an ActivMedia Pioneer 3-DX. The mirror was a large wide-view hyperbolic mirror from Accowle Ltd.

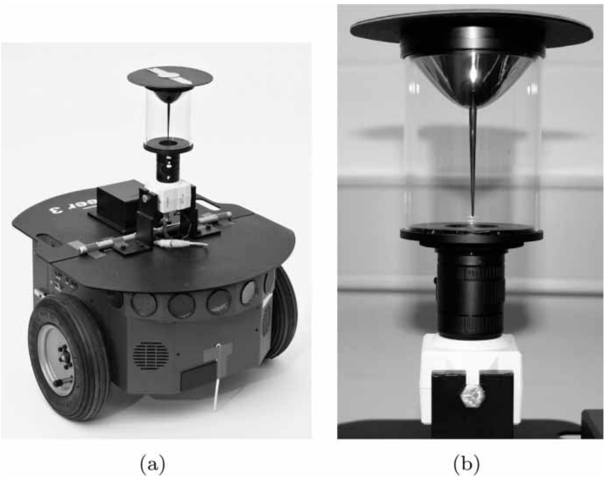


Figure 1. (a) The ActivMedia Pioneer 3-DX robot with attached camera and hyperbolic mirror. (b) A close-up of the camera and hyperbolic mirror assembly.

were then low-pass filtered (butterworth filter: relative width 0.25, order 1) and downsampled to a relatively low resolution (206 × 46). This level of resolution is roughly consistent with that of honeybee eyes.[†]

Images were collected with the room in a number of different conditions. The names and descriptions of these image collections are as follows.

original The standard or default condition of the room. Images were collected with both overhead fluorescent light bars on, the curtains and door closed, and no extraneous objects present.

chairs Three additional office chairs were positioned within the capture grid.

arboreal A tall (3 m) indoor plant was added to the centre of the capture grid.

screen A projection screen (height 2.55 m) was added to the centre of the capture grid. The screen ran parallel to the long-axis of the room.

day Images were collected with curtains open in full daylight.

twilight Curtains and door open. Image collection began at 18:34 on 13 March 2004 just after sunset.[‡] At this time, the room was still receiving plenty of daylight. The last image was captured at 19:14. The sky outside was quite dark by this time.

doorlit The light bar near the window was switched off.

winlit The light bar near the door was switched off.

Figure 2 shows sample images from all of the above collections. Henceforth, we shall refer to images from these collections as being from the *office environment*.

Images were also collected from two locations within the main hall of Bielefeld University. These locations were categorically different from the office environment in that the distances to viewed objects ranged from several metres to hundreds of metres. These image collections

[†] Land (1997) reviews the visual acuity data of many insects and presents four differing estimates of *Apis mellifera* (honeybee) interommatidial angles. Larger interommatidial angles give lower resolution images. The low resolution used here is approximately consistent with the largest interommatidial angle reported by Land ($\Delta\phi = 1.7°$).

[‡] The official time of sunset on that day was 18:32 for Osnabrück Airport, approximately 50 km away.

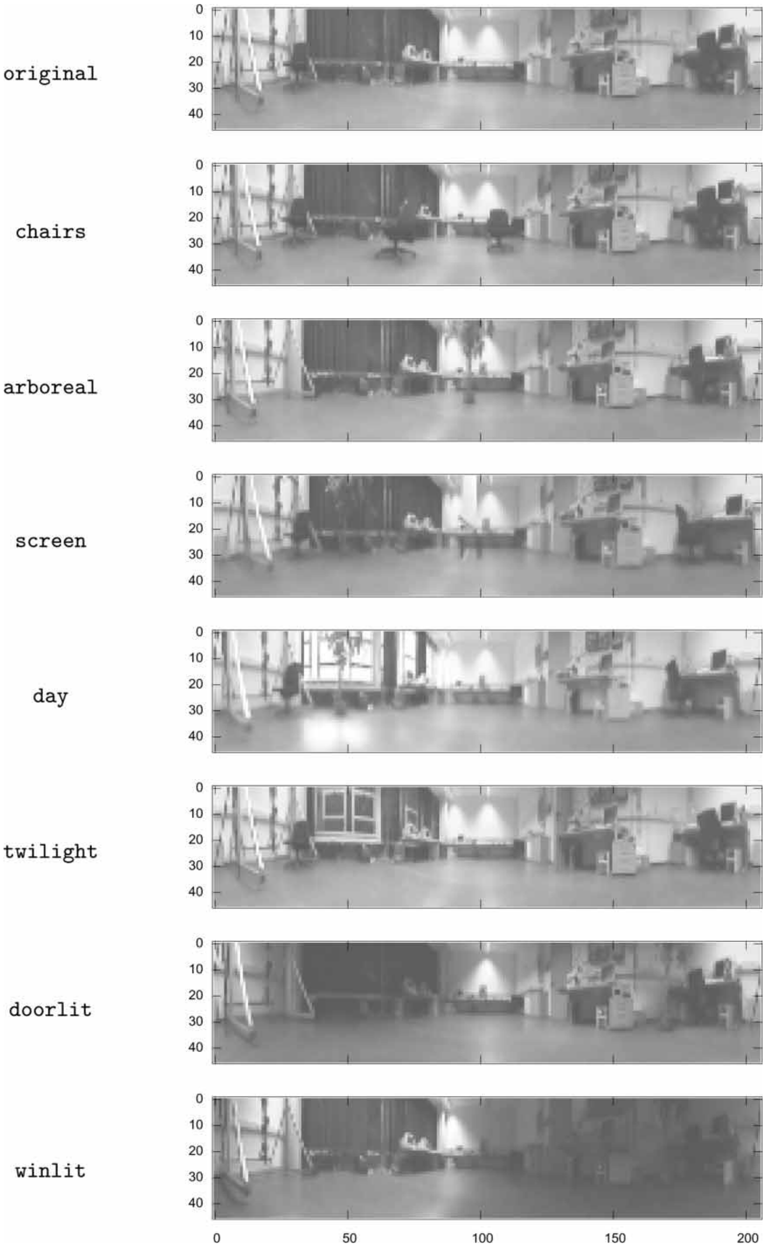


Figure 2. Sample images from eight image collections in the office environment. Images taken from position (3,0).

are larger, spanning 4.5 × 10 m in area at 50 cm resolution. All images were captured late at night when the hall was free of people. Two sets of images were collected:

hall1 Collection area was adjacent to a dining area with two large open areas extending off in opposite directions and the large main hall extending out opposite the dining area. Overhead structures are visible and appear in significantly different positions within the collected images.

hall2 Collection area was near a message board at approximately the centre of the hall.

Figure 3 shows sample images from these two collections.

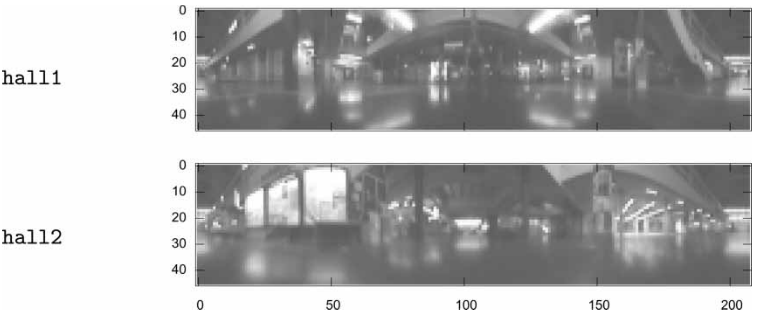


Figure 3. Sample images from two image collections in the hall environment. Images taken from position (5,10).

This image database is available for public download at http://www.ti.uni-bielefeld.de/html/research/avardy.

## 3.2 *Assumptions*

**3.2.1 Orientation.** The images from the database described above were all taken from approximately the same orientation. Thus, in so far as the experiments on this database emulate the behaviour of a robot, it is assumed that this robot has some means of measuring its orientation. Borenstein *et al.* (1996) discussed a range of techniques and devices for accurately estimating orientation. Orientation can also be recovered through visual means alone (Röfer 1997, Gourichon *et al.* 2003, Zeil *et al.* 2003). Franz *et al.*'s warping method includes orientation in its search space and therefore has the advantage of not requiring a compass.

Meanwhile, it appears that insects such as honeybees do require a compass in order to home visually (Cartwright and Collett 1983). Bees and ants are able to infer orientation from the polarization pattern of light in the sky (Wehner *et al.* 1996), as well as by detecting the local magnetic field (Collett and Baron 1994). While homing, it is therefore possible for them to restore their orientation to that held when capturing the snapshot image. This might be achieved only for particular moments during homing, or throughout a homing episode. Bees are known to take up characteristic orientations when close to a goal, such as facing magnetic south (Collett and Baron 1994). This would free them from the requirement of associating particular orientations with particular goals. All of the above suggests that the visual homing methods used by insects are based on the same assumption as made here – that the orientation held at the goal is recovered to support homing.

**3.2.2 Normalised home vectors.** The methods developed in this paper generate unit-length home vectors. This means that only the direction to the goal is computed, not the distance. However, it can be advantageous to estimate the distance to the goal and use this information to take smaller steps when close to the goal and larger steps otherwise. Franz *et al.*'s warping method approximates this distance as part of its regular operation. However, our implementation of the warping method generates unit-length home vectors. Computing the correct direction of the goal is a more fundamental ability than estimating its direction. The ability to estimate distance could be incorporated into our methods through the relatively simple techniques of Röfer (1997) or Zeil *et al.* (2003).

### 3.3 *Performance metrics*

To assess a homing method on a particular image collection, the following procedure is used. A single image from the image collection is selected as the snapshot image. The coordinates of this image within the capture grid are taken as the goal coordinates. The snapshot image is then input to the homing method. All remaining images are then input, in turn, to the homing method. For each of these images the homing method will return a home vector. The home vector fields produced in this manner are assessed by two metrics: the *average angular error* and the *return ratio*.

**3.3.1 Average angular error.** Perhaps the simplest means of measuring homing performance is to compute the *average angular error* between observed home vectors and true home vectors. For analysis purposes, we know the coordinates of both the current position **c**, and the goal position **s**. Therefore, $\mathbf{v}_{\mathsf{Ideal}}$ is known

$$\mathbf{v}_{\mathsf{Ideal}} = \frac{\mathbf{s}-\mathbf{c}}{\|\mathbf{s}-\mathbf{c}\|}. \tag{1}$$

Given a homing method, ‘Method’, which computes a unit-length home vector $\mathbf{v}_{\mathsf{Method}}$, the angular error ($AE$) is defined as

$$AE(\mathbf{s},\mathbf{c}) = \arccos(\mathbf{v}_{\mathsf{Ideal}}^{\mathrm{T}}(\mathbf{s},\mathbf{c})\mathbf{v}_{\mathsf{Method}}(\mathbf{s},\mathbf{c})), \tag{2}$$

measured in radians.

The average angular error ($AAE$) is computed for a single snapshot position **s** by finding the average value of equation (2) over all other positions. Assume we have an image collection of width $m$ and height $n$, and that position coordinates are the same as the indices of the $m \times n$ capture grid. $AAE$ for snapshot position **s** is defined as

$$AAE_{\mathbf{s}} = \frac{1}{mn-1}\sum_{c_x=0}^{m-1}\sum_{c_y=0}^{n-1} AE(\mathbf{s},\mathbf{c}), \tag{3}$$

likewise measured in radians. Note that the angular error at the snapshot position is excluded from this summation. $AAE$ is a useful performance metric because it is easy both to understand and to compute. It has also been employed by other robot homing studies (Weber *et al.* 1999, Hafner 2001).

**3.3.2 Return ratio.** In the biological literature, homing success has been described in terms of *catchment area* (Cartwright and Collett 1987). The catchment area is the area surrounding the goal within which the agent is able to home successfully. Average angular error is an indirect measure of catchment area. For a perfect home vector field with zero angular error, the catchment area covers the entire capture grid. However, while average angular error is invariant to the location of errors, the size of the catchment area will depend very much on where errors occur.

To account for the position of errors and to provide a more direct measurement of catchment area, a simulation of homing is performed. A dummy agent is placed at all integer grid positions and allowed to move according to the home vectors which have been pre-computed for the method in question. The agent moves in steps of length $l_{\mathrm{rr}}$ (set to 0.5 here, see Vardy (2005) for explanation). The home vector for the integer grid position closest to the agent’s current position dictates the agent's movement. The outcome of the homing attempt is binary. If the agent reaches within a distance of one to the goal position then the attempt is considered to be successful. The total distance that the agent is allowed to travel is set to half the circumference of the capture grid. For image collections from the office environment this is $9 + 16 = 25$. If the agent travels this distance but has not yet reached the goal then the attempt is considered to be unsuccessful. This technique is similar to those described by other robot visual homing studies (Franz *et al.* 1998b, Stürzl and Mallot 2002).

The output from the above procedure is a binary map of successful versus unsuccessful homing attempts for all start positions. The *return ratio* is computed by counting the number of successful positions and dividing by the total number of homing attempts ($10 \times 17 - 1 = 169$ for the office environment). We indicate the return ratio for snapshot position $(x, y)$ as $RR_{(x,y)}$. This quantity is in the range [0, 1], where 1 indicates perfect homing.

### 3.4 *Performance across snapshot positions*

Homing performance is dependent on the chosen snapshot position. To assess this dependence we test homing methods on a sampling of 20 snapshot positions. Figure 4 shows these snapshot positions, which were chosen to sample the capture grid evenly.

Performance metrics *AAE* and *RR* are each computed for all 20 snapshot positions. To obtain an overall measure of homing performance according to these two metrics, the metric's value is averaged across all snapshot positions. This is indicated as either $AAE_*$ or $RR_*$. To specify that such a value was obtained on a particular image collection, this collection is specified (e.g. $AAE_*$(original)).

### 3.5 *Statistical tests*

Observed differences in the performance metrics might be due to chance. Statistical tests are performed to analyse the probability of such an occurrence. For the *AAE* metric a repeated measures analysis of variance (ANOVA) test is used to compare differences among homing methods (Zar 1996). The individual subjects of the test are the *AAE* values for each

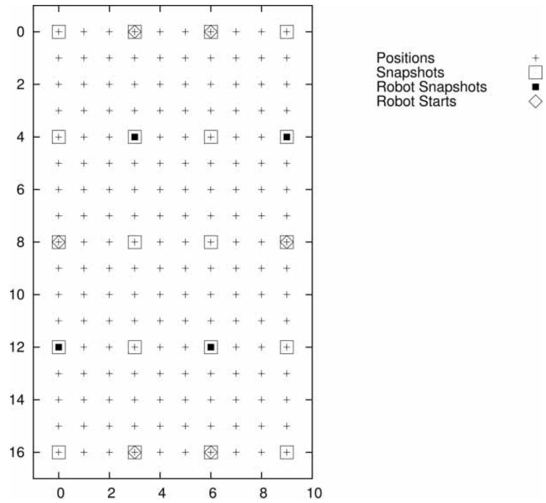


Figure 4. Positions of images, snapshots, robot snapshot positions and robot start positions within the office environment capture grid.

snapshot position. As the same subjects (snapshot positions) are used for all treatments (homing methods), a repeated measures design is required. If the null hypothesis (equal *AAE* across methods) is rejected then the Tukey–Kramer multiple comparisons test is carried out to determine the significance of differences between methods. This ANOVA assumes that the distribution of *AAE* is well approximated as normal. However, the distribution of *RR* is quite different from a normal distribution. Therefore, to test the difference in *RR* between homing methods, Friedman's test is used (Zar 1996). This test is essentially the non-parametric equivalent of the repeated measures ANOVA. Dunn's post-test is used to determine the significance of differences between methods should the null hypothesis (equal *RR* across methods) be rejected. All statistical tests were carried out using GraphPad's InStat package.[†]

### 3.6 *Robot homing trials*

To provide further justification for the results presented above, a series of online homing trials was carried out. In these trials the computed home vector directed the movement of the Pioneer robot. A subsampling of positions on the capture grid was used for the snapshot and start positions. Figure 4 shows both the four snapshot positions tested and the six positions from which homing trials were started. As stated earlier, we assume the existence of a compass to allow the orientation held at the snapshot position to be recovered. We found that a magnetic compass was not reliable within the office environment. Therefore, we used the robot's built-in odometry system to obtain an estimate of the angular deviation between the snapshot position and the current view. The current view was then rotated by this amount and the computed home vector likewise rotated by the negative of this angle. When positioning the robot to capture a snapshot image, the robot was manually oriented by aligning the mounted laser pointers with tape marks along the wall. At the start of each homing attempt the robot was manually oriented in the same fashion and the odometry compass reset.

All image processing and other operations were carried out on the robot's onboard Pentium III processor. The robot would process the current image, compute a home vector and then rotate to align itself with this vector. It would then travel 50 cm in a straight line, stop, and then capture another image. The robot was allowed 15 moves to reach home. The warping method occasionally detected that the home position had been reached by noting that the unwarped current image was more similar to the snapshot image than all of the warped images. In this case, it would output a home vector of length zero and the homing attempt would end. Our method was not equipped with any mechanism to recognize that the goal had been reached. Instead, it would reach the goal and begin a series of oscillations around it until all 15 moves had been carried out. The Pioneer was equipped with a ring of sonar sensors, which were used to test for imminent collision. If the robot approached within a threshold distance of any object it would halt and the homing attempt would end.

Each time the Pioneer captured an image, a separate observation camera, positioned above the Pioneer and looking down on it, would capture an image of the Pioneer. After all homing trials were complete, the position of the Pioneer was tracked by manually selecting it in images from the observation camera. The homography matrix relating the image coordinates of the Pioneer with the coordinate system used for the image database was then calculated using the normalized direct linear transformation algorithm (Hartley and Zisserman 2004), as implemented by Kovesi.[‡]

[†] See http://www.graphpad.com.

[‡] See http://www.csse.uwa.edu.au/∼pk/Research/MatlabFns/.

## 4. Homing by block matching

### 4.1 *Computation of correspondence vectors*

Assume we have a snapshot image $S$, a current image $C$, and we wish to find correspondences between these two images. We further assume that these two images were captured at the same compass orientation. Correct correspondence is achieved if for each origin point $\mathbf{p}$ in $S$ we can find an appropriate candidate point $\mathbf{p}'$ in $C$ such that the same image features appear in the local regions surrounding these two points. The method 'BlockMatch' operates by doing a search for each $\mathbf{p}$ in $S$ to find the best matching region in $C$. The quality of a match is judged by calculating the SSD

$$\mathrm{SSD}(\mathbf{p}, \mathbf{p}') = \sum_{i=-r}^{r} \sum_{j=-r}^{r} [S(p_x + i, p_y + j) - C(p'_x + i, p'_y + j)]^2, \tag{4}$$

with $i, j \in \mathbb{Z}$. A low value of SSD indicates a good match. The quantity $r$ is the block radius. Large values of $r$ require regions to match over a large area and have increased computational cost. Therefore, we use $r = 1$.

A correspondence search is carried out by BlockMatch not for every position in $S$ but only for a sampling of positions at integer multiples of $m_x$ in the horizontal direction and $m_y$ in the vertical direction, where $m_x$ and $m_y$ are also integers. For images of width $w$ and height $h$, the number of horizontal and vertical sampling points is as follows:

$$n_x = \lfloor w/m_x \rfloor \tag{5}$$

$$n_y = \lfloor h/m_y \rfloor. \tag{6}$$

Thus, the total number of correspondence searches is $n_x n_y$. Each search involves computing the SSD for all positions in $C$ within a radius of $q$ pixels of $\mathbf{p}$, and selecting $\check{\mathbf{p}}$ as the candidate match with the lowest SSD:

$$\check{\mathbf{p}} = \arg \min_{\mathbf{p}' \in E_q(\mathbf{p})} \mathrm{SSD}(\mathbf{p}, \mathbf{p}') \tag{7}$$

$$E_q((p_x, p_y)) = \{(p_x + i, p_y + j) | i, j \in \mathbb{Z}, |i| \leq q \wedge |j| \leq q\}. \tag{8}$$

We add the constraint that origin points $\mathbf{p}$ in the snapshot image will only be paired with candidate points in the current image $\mathbf{p}'$ that are on the same side of the horizon. The horizon of the panoramic image is the line that does not undergo vertical translations under movements of the robot in the plane. As long as the robot moves purely within a single plane, no image features should cross the horizon. Therefore, we constrain our search to avoid any such spurious pairings.

The position $\check{\mathbf{p}}$ with the lowest SSD is used to compute the correspondence vector $\boldsymbol{\delta}$

$$\boldsymbol{\delta} = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \Delta_x(\check{p}_x - p_x) \\ \Delta_y(\check{p}_y - p_y) \end{pmatrix}, \tag{9}$$

where $\Delta_x$ represents the inter-pixel angle in the horizontal direction and $\Delta_y$ represents vertical inter-pixel angle. The factors $\Delta_x$ and $\Delta_y$ imply that $\boldsymbol{\delta}$ is expressed as a pair of angles as opposed to image indices.

Each correspondence vector ideally describes the movement of an image feature in the snapshot image to its new position in the current image. From each of these correspondence vectors we can determine an individual home vector. For an agent travelling within a plane, there is a one-to-one mapping between correspondence vectors and home vectors, which will be described in the next section.

## 4.2 *Vector mapping*

Figure 5(a) shows a top-down view of an agent at its current position **c**, the goal position **s**, and a landmark **l** visible from both positions. We assume the agent's orientation at **c** is the same as at **s**. The translation from **s** to **c** occurred at an angle $\alpha$ from the agent's heading (indicated by the arrow). At **s**, the angular position of **l** in the agent's field of view was $\theta_x$. At **c**, the view of

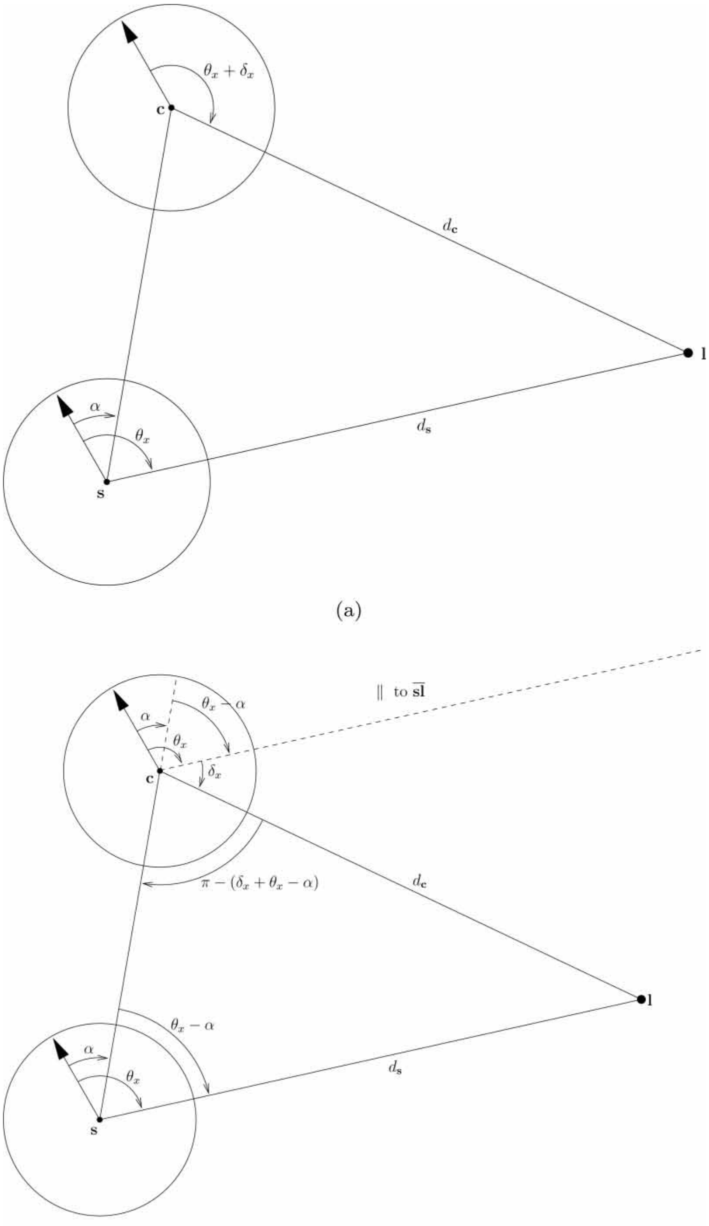


Figure 5. Motion of an agent from **s** to **c** and the resultant change in viewing position of a landmark **l**.

**l** has shifted by $\delta_x$. The distances $d_\mathbf{s}$ from **s** to **l**, and $d_\mathbf{c}$ from **c** to **l** are included in the figure, however both are unknown.

The correct direction home is given by $\beta = \alpha + \pi$. We wish to compute $\beta$ given the correspondence vector $\boldsymbol{\delta}$ and the projection of **l** in the snapshot image, given by $(\theta_x, \theta_y)$. Unfortunately, with the situation described thus far, the lengths of all three sides of triangle $\Delta$**lsc** remain unknown. If we have no information about these lengths then it becomes impossible to compute $\beta$ exactly. As a means of circumventing this dilemma, Franz *et al.* (1998b) proposed two assumptions about the distribution of landmarks in the environment that allow $\beta$ to be approximated. However, no such assumptions are required if we can just determine the ratio between the distances $d_\mathbf{s}$ and $d_\mathbf{c}$. This ratio can be obtained by considering the vertical components $\delta_y$ and $\theta_y$.

Figure 6 shows **s**, **c** and **l** when viewed from an oblique angle above the plane containing **s** and **c**. The agent is constrained to travel such that the centre of its imaging sphere lies within the plane (and thus the plane projects on the image horizon). Landmark **l** lies a perpendicular distance $h_\mathbf{l}$ above (or below) the plane. This picture and the use of the vertical components $\delta_y$ and $\theta_y$ allow us to calculate the ratio $d_\mathbf{s}/d_\mathbf{c}$. First we find the tangents for the two angles of **l** above the plane.

$$\tan\theta_y = \frac{h_\mathbf{l}}{d_\mathbf{s}} \tag{10}$$

$$\tan(\theta_y + \delta_y) = \frac{h_\mathbf{l}}{d_\mathbf{c}}. \tag{11}$$

We now eliminate $h_\mathbf{l}$ to obtain the desired ratio

$$\frac{d_\mathbf{s}}{d_\mathbf{c}} = \frac{\tan(\theta_y + \delta_y)}{\tan\theta_y}. \tag{12}$$

Note that equation (12) becomes numerically unstable when $\theta_y$ is close to zero. This makes intuitive sense. We cannot reliably compute the ratio $d_\mathbf{s}/d_\mathbf{c}$ if the landmark **l** lies on the horizon. In the implementation of this vector mapping technique, points very near the horizon are omitted ($|\theta_y| \leq 0.01$).

Knowing how to compute the ratio $d_\mathbf{s}/d_\mathbf{c}$, we return to the analysis of horizontal angles with the goal of deriving an expression for $\beta$. Figure 5(b) is identical to figure 5(a) except that it shows a number of intermediary angles that will be useful below. We begin by determining the interior angles of the triangle at **s** and **c**. $\angle$**s** is just $\theta_x - \alpha$. To determine $\angle$**c** it is helpful to

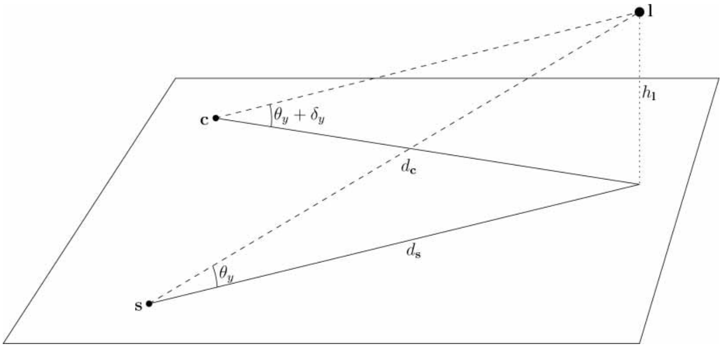


Figure 6. Oblique view of plane containing **s** and **c**. Solid lines are within the plane, dashed lines are above and oblique to the plane, and the dotted line is perpendicular to the plane.

extend the line $\overline{\mathbf{sc}}$ and to add a line parallel to $\overline{\mathbf{sl}}$ and position it through $\mathbf{c}$. These constructions are shown as dashed lines in figure 5(b). We obtain

$$\angle\mathbf{c} = \pi - (\delta_x + \delta_x - \alpha). \tag{13}$$

The law of sines can now be applied

$$\frac{\sin(\pi - (\delta_x + \theta_x - \alpha))}{d_\mathbf{s}} = \frac{\sin(0_x - \alpha)}{d_\mathbf{c}}. \tag{14}$$

Appendix section B.1 continues the development of equation (14). Via some simple algebra and the use of some trigonometric identities, we arrive at the following expression:

$$\beta = \theta_x - \arctan\left(\frac{\sin\delta_x}{\dfrac{\tan(\theta_y + \delta_y)}{\tan\theta_y} - \cos\delta_x}\right) + \pi. \tag{15}$$

We transform the angle $\beta$ calculated for each correspondence vector into a home vector $\mathbf{u}$

$$\mathbf{u}(\theta_x, \theta_y) = \begin{pmatrix}\cos\beta\\ \sin\beta\end{pmatrix}. \tag{16}$$

### 4.3 *Computing the final home vector*

If the correspondence vector $\boldsymbol{\delta}$ is correct then the vector $\mathbf{u}$ represents the true home vector. Thus, if it was known that all correspondence vectors were correct, we could simply map any one of these into a home vector and return that home vector as our final answer. However, finding correct correspondences is non-trivial and we assume that some correspondence vectors are incorrect. One possibility would be to use robust estimation techniques such as the RANSAC algorithm to select a maximally consistent set of correspondence vectors (Fischler and Bolles 1981).

A much simpler strategy is to sum all individual home vectors together to produce a final home vector. This strategy has the added benefit of reducing the impact of incorrect correspondences. Correct correspondence vectors will generate correct individual home vectors – all pointing in approximately the same direction. We shall refer to incorrect correspondence vectors as *mismatches*. If the mismatches are uncorrelated then the individual home vectors generated from them will also be uncorrelated. Thus, home vectors computed from mismatches will tend to cancel each other out in the summation. The main contribution to the final home vector will then be from the correct individual home vectors. Hence, the final home vector will be approximately correct. We refer to the assumption that incorrect correspondences are uncorrelated as the *uncorrelated mismatches assumption*.

Given this assumption, we compute a total home vector $\mathbf{v}$ by summing all individual home vectors $\mathbf{u}$ together. Finally, $\mathbf{v}$ is normalized, $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$. The vector $\hat{\mathbf{v}}$ is the final home vector.

### 4.4 *Pre-processing*

In its typical application within computer vision, block matching is applied directly without any image pre-processing. In the parameter search described below the option of no pre-processing was available. Also available was the use of a Sobel filter to enhance edges and the use of a Gaussian low-pass filter.[†] The number of applications of the Gaussian filter was left as a parameter.

Also, the top and bottom portions of the image tend to exhibit less distinguishing features than regions near the horizon. Therefore, one of the parameters below controls how many of the top and bottom rows to exclude from the matching process.

### 4.5 *Optimizing parameters*

The approach taken throughout this paper for all homing methods is to optimize performance on the original image collection by doing a search for best parameters. Each combination of parameters is judged by configuring the homing method with those parameters and then determining $AAE_*$ (original). The combination with the lowest value is then used for subsequent experiments. Thus, original is the 'home ground' for all methods.

The parameters tested for BlockMatch are as follows:

- Number of applications of Gaussian filter: **0**, 1, 2, 3, or 4.
- Use Sobel filter: true or **false**.
- Number of top and bottom image rows to exclude: 0, 5, or **10**.
- Search radius, $q$: 5, 10, **30**, or 50.

The best parameters found are given above in bold. The step size parameters of $m_x$ and $m_y$ were not varied in this search but were set to a moderate value of $m_x = m_y = 4$.

### 4.6 *Optimizing parameters for Warping*

The parameters tested for Warping are:

- Angular distance above and below the horizon defining the region of the two-dimensional image used to create one-dimensional images: **5**°, 10°, or the whole image (23°).
- Downsampling factor for one-dimensional images, $m_x$: **1** (no downsampling), **2**, or 4.
- Number of applications of a Gaussian low-pass filter: 0, **1**, 3 or 5.
- Number of search increments for direction, $n_\alpha$: 32 or **64**.
- Number of search increments for distance, $n_\rho$: 32 or **64**.
- Number of search increments for rotation, $n_\psi$: **1**, 32 or 64.
- Method of comparison between warped and snapshot images: **sum of squared differences** or dot product.

The best combination found for Warping is indicated in bold above. Generally, the performance of this method is relatively robust to most of these parameters (as mentioned by Möller (2002a)). Interestingly, the best combination has $n_\psi = 1$, meaning that no search is done for rotation. This makes sense as the images in the database were all taken from the same orientation. The mere fact that the best combination has $n_\psi = 1$ means that adding rotation as an extra dimension of search can impair performance.

[†] This Gaussian operator convolves the image by the kernel

$$[0.005\ 0.061\ 0.242\ 0.383\ 0.242\ 0.061\ 0.005]$$

applied separately in the $x$ and $y$ directions. Where the kernel overlaps the top or bottom boundary of the image the output pixel value will be set to a special out-of-bounds value, effectively shrinking the image vertically.

## 4.7 *Results on image database*

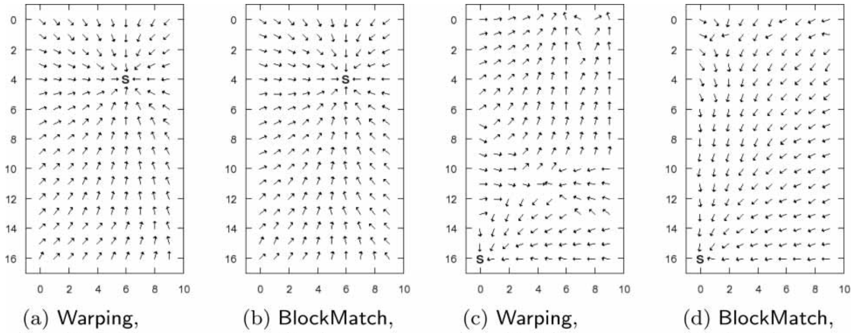
**4.7.1 Image collection original.** Figure 7 shows the home vectors generated by Warping and BlockMatch on image collection original for snapshot positions (6,4) and (0,16). At position (6,4), the home vector fields for both methods are approximately correct. That is, all vectors point towards the snapshot position with relatively little error. The captions in this figure indicate the value of *AAE* and *RR* computed for these vector fields. Both fields for position (6,4) show relatively low *AAE* (much lower than $\pi/2$) and a perfect value of one for *RR*. At position (0,16), the home vector field for BlockMatch is also approximately correct. However, the field for Warping is approximately correct for only a small region surrounding the goal. As measured by the *RR* metric, the catchment area for Warping at position (0,16) covers only 26.6% of the capture grid.

Figure 8 shows the computed *AAE* and *RR* for Warping and BlockMatch for the 20 snapshot positions depicted in figure 4. For both performance metrics, BlockMatch performs extraordinarily well. In fact, for the *RR* metric it performs perfectly – achieving a return ratio of one for every snapshot position. Warping exhibits low *AAE* and high *RR* throughout most of the capture grid, but exhibits poor performance along the boundaries of the capture grid, particularly in the corners.

It makes sense that Warping should perform poorly in these regions of the capture grid. The boundaries of the capture grid are adjacent to the walls, desks and chairs of the office environment. Thus, the equal distance assumption that Warping relies upon will be violated most strongly for snapshot positions near these objects. Still, Franz *et al.* (1998b) argue that even when this assumption is violated, the error remains small as long as the agent's distance, $b$, from the goal is less than the distance from the goal to the nearest landmark $d_s$. Yet, if we are at the boundary, then $d_s$ will be small. Thus, the inequality $b < d_s$ will be satisfied only for very small displacements.

Also, the region at the bottom of the capture grid is problematic for another reason. At the end of the room that this region represents there is a large bench against a white wall. Images taken here show the bench as large and dark. From further out, however, the white wall is more prevalent. This may also contribute to Warping's poor performance in the bottom left corner of the capture grid.

**4.7.2 Image Collections hall1 and hall2.** Figure 9 shows home vectors generated by Warping and BlockMatch on the hall image collections at snapshot position (3,12).



(a) Warping, $\mathbf{s} = (6, 4)$: $AAE = 0.119$, $RR = 1.000$

(b) BlockMatch, $\mathbf{s} = (6, 4)$: $AAE = 0.165$, $RR = 1.000$

(c) Warping, $\mathbf{s} = (0, 16)$: $AAE = 1.758$, $RR = 0.266$

(d) BlockMatch, $\mathbf{s} = (0, 16)$: $AAE = 0.196$, $RR = 1.000$

Figure 7. Home vector fields for (a, c) Warping and (b, d) BlockMatch, for snapshot positions (a, b) (6,4) and (c, d) (0,16) in image collection original.

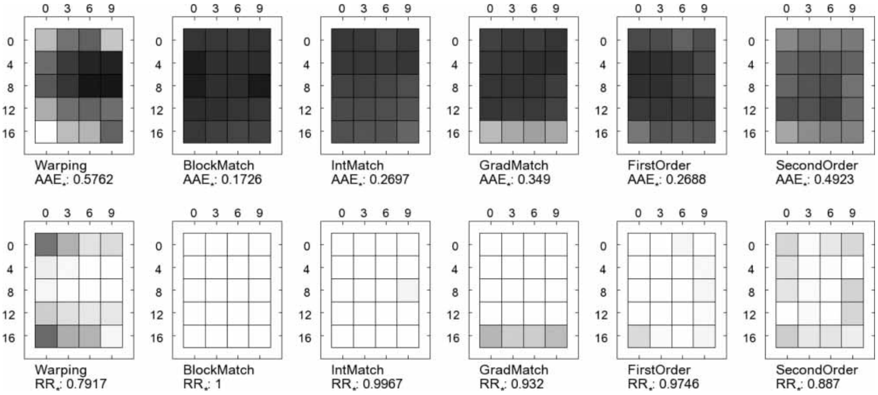
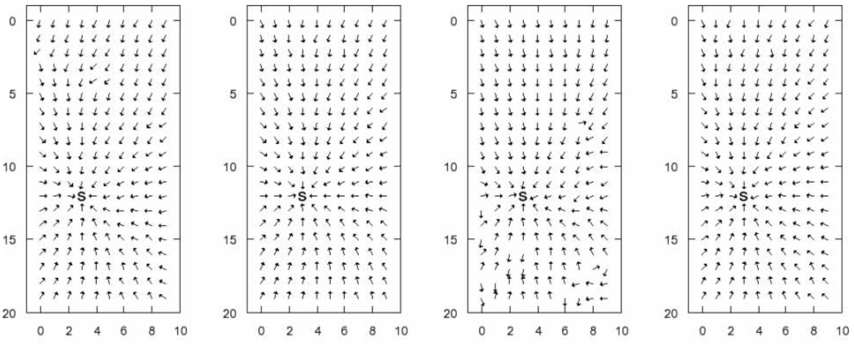


Figure 8. *AAE* and *RR* across snapshot positions for all methods on image collection original. *AAE* max. = 1.758, *RR* max. = 1.

Both methods yield approximately correct home vector fields. However, there are a number of incorrect vectors embedded in the vector field for Warping in hall2. Errors in BlockMatch's vector field remain relatively low.

Figure 10 shows the computed *AAE* and *RR* across all 20 snapshot positions for both methods on image collection hall1. Again, BlockMatch's performance remains quite high across snapshot positions. For the *AAE* metric, Warping exhibits relatively low error for most snapshot positions. For the *RR* metric, Warping performs well except for a few particular snapshot positions. Figure 11 shows similar results for image collection hall2.

**4.7.3 Cross-collection tests.** In addition to original, seven other image collections were captured in the office environment. These image collections allow us to determine not only how each method performs within a single environment, but also how each performs when snapshot images are taken from one environment and current images taken from another. In this way, we can determine the robustness of performance to a changing environment. As insects are manifestly able to handle changes in lighting and the movement of objects (such as



(a) Warping, hall1, $\mathbf{s} = (3, 12)$: $AAE = 0.160$, $RR = 1.000$

(b) BlockMatch, hall1, $\mathbf{s} = (3, 12)$: $AAE = 0.144$, $RR = 1.000$

(c) Warping, hall2, $\mathbf{s} = (3, 12)$: $AAE = 0.407$, $RR = 0.915$

(d) BlockMatch, hall2, $\mathbf{s} = (3, 12)$: $AAE = 0.114$, $RR = 1.000$

Figure 9. Home vector fields for (a, b) Warping and (c, d) BlockMatch, for snapshot position (3, 12) in image collections hall1 (a, c) and hall2 (b, d).

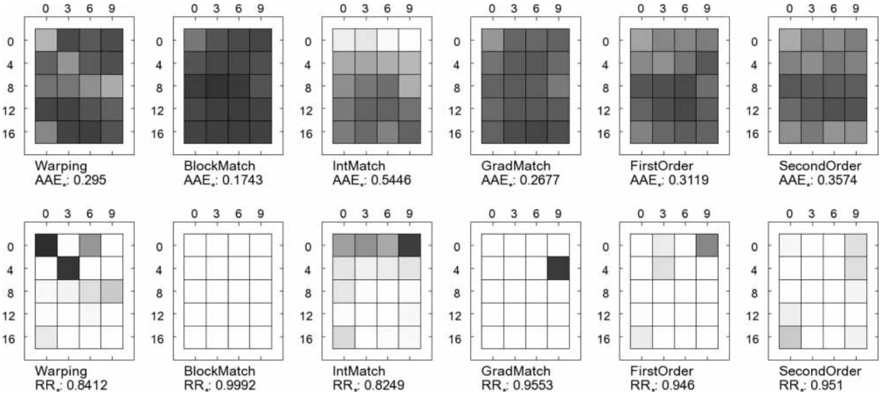


Figure 10. *AAE* and *RR* across snapshot positions for all methods on image collection hall1. *AAE* max. = 2.288, *RR* max. = 1.

wind-blown vegetation), so must biorobotic models of insect homing be capable of handling such phenomena. While there are certainly differences in the dynamic events that can occur in an office environment as opposed to an outdoor setting, it is apparent that homing performance should survive as many such events as possible.

For the eight office environment image collections, there are 64 pairings. Each pairing involves taking 20 snapshot images from one collection and allowing the method in question to compute home vectors for all images from the other collection. We compute $AAE\mathbf{s}(\mathbf{a}, \mathbf{b})$ for each snapshot position **s**, and average these over all 20 snapshots to obtain $AAE_*(\mathbf{a}, \mathbf{b})$, where **a** is the collection from which snapshot images are drawn and **b** is the collection from which current images are drawn. Similarly, we obtain $RR_*(\mathbf{a}, \mathbf{b})$. To obtain an overall measure of homing performance we average $AAE_*$ over all pairings of office environment image collections to obtain the *cross-collection angular error* $AAE_*(*, *)$. The average of $RR_*$ over all pairings yields the *cross-collection return ratio* $RR_*(*, *)$.

Figure 12 plots $AAE_*$ and $RR_*$ for all such pairings for both Warping and BlockMatch. Also shown in the figure are the cross-collection angular error and return ratio metrics. It is apparent that both methods are robust to many environmental modifications. Pairings with the doorlit and winlit image collections appear to be the most difficult for both. This is due to the very large change in illumination caused by turning off one of the room's light bars. Pairings with

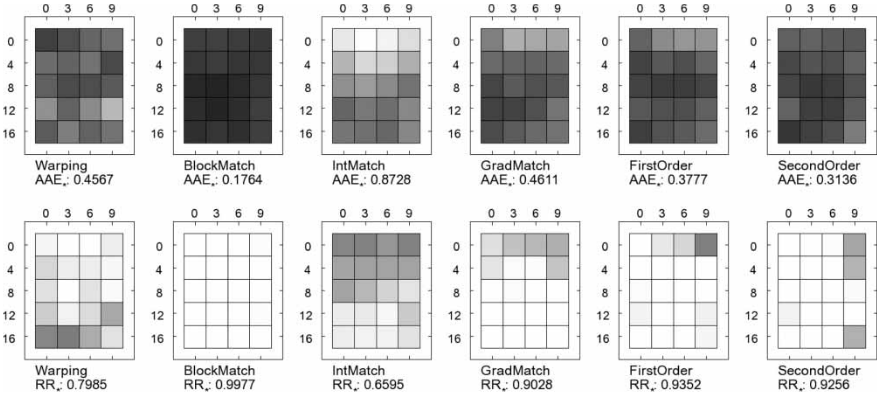


Figure 11. *AAE* and *RR* across snapshot positions for all methods on image collection hall2. *AAE* max. = 2.127, *RR* max. = 1.

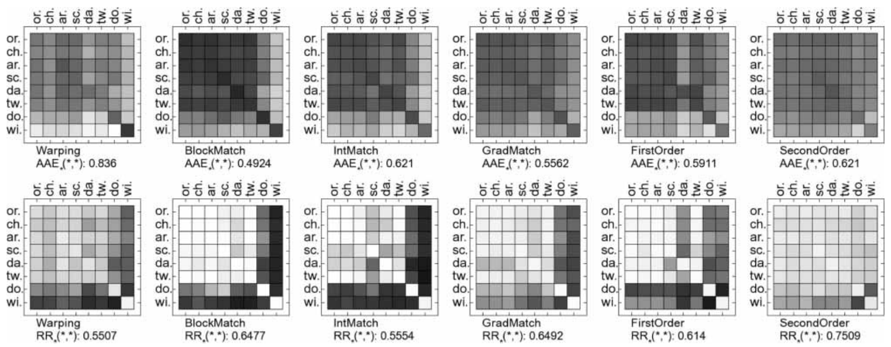


Figure 12. $AAE_*$ and $RR_*$ over all pairings of the eight office image collections. The image collection used for snapshot images is shown on the vertical axis, while the collection used for current images is shown on the horizontal axis. Abbreviations: or., original; ch., chairs; ar., arboreal; sc., screen; da., day; tw., twilight; do., doorlit; wi., winlit. $AAE$ max., = 1.811, $RR$max. = 1.

the winlit collection are particularly difficult because the side of the room that remains lit is also the most featureless. Interestingly, when winlit is paired with winlit, Warping exhibits its lowest $AAE_*$ and its highest $RR_*$. A room that is half in light and half in shadow can be easily mapped to one-dimensional images that are quite useful for navigation. Overall, BlockMatch appears to be more robust to environmental modifications than Warping. This is confirmed by the statistical tests below.

## 4.8 *Comparative results*

The statistical tests described in section 3.5 were carried out to compare the performance of Warping and all five methods presented in this paper against each other. In this section, we consider only differences between Warping and BlockMatch.

Table 1 presents the results of these tests on image collection original. The top part of the table shows that BlockMatch exhibits significantly lower $AAE$ than Warping. A complication

Table 1. Statistical significance of the difference in $AAE_*$ (original) and $RR_*$(original) between homing methods.†

| | Warping | BlockMatch | IntMatch | GradMatch | FirstOrder | SecondOrder |
|---|---|---|---|---|---|---|
| *Significance of difference in $AAE_*$(original)* | | | | | | |
| Warping | X | | | | | |
| BlockMatch | $***$ | X | | | | $***$ |
| IntMatch | $***$ | | X | | | $**$ |
| GradMatch | $**$ | | | X | | |
| FirstOrder | $***$ | | | | X | $**$ |
| SecondOrder | | | | | | X |
| *Significance of difference in $RR_*$(original)* | | | | | | |
| Warping | X | NA | | | | |
| BlockMatch | NA | NA | NA | NA | NA | NA |
| IntMatch | $***$ | NA | X | | | $***$ |
| GradMatch | $**$ | NA | | X | | $*$ |
| FirstOrder | $*$ | NA | | | X | $*$ |
| SecondOrder | | NA | | | | X |

†Significance for each cell is indicated if the method on the vertical axis is significantly better (lower $AAE$ or higher $RR$) than the method on the horizontal axis. Empty fields indicate no significant difference. For $AAE$, a repeated measure ANOVA with Tukey–Kramer post-test was used. For $RR$ Friedman's test with Dunn's post-test was used. $*p < 0.05$; $**\ p < 0.01$; $***p < 0.001$; $****\ p < 0.0001$; X, self-match; NA, not applicable (see text).

Table 2. Statistical significance of the difference in $AAE_*$ (hall1) and $RR_*$(hall1) between homing methods; see table 1 for notation.

| | Warping | BlockMatch | IntMatch | GradMatch | FirstOrder | SecondOrder |
|---|---|---|---|---|---|---|
| *Significance of difference in* $AAE_*$*(hall1)* | | | | | | |
| Warping | X | | ∗∗∗ | | | |
| BlockMatch | ∗ | X | ∗∗∗ | | ∗∗ | ∗∗∗ |
| IntMatch | | | X | | | |
| GradMatch | | | ∗∗∗ | X | | |
| FirstOrder | | | ∗∗∗ | | X | |
| SecondOrder | | | ∗∗∗ | | | X |
| *Significance of difference in* $RR_*$*(hall1)* | | | | | | |
| Warping | X | | | | | |
| BlockMatch | ∗ | X | ∗∗ | | | |
| IntMatch | | | X | | | |
| GradMatch | | | ∗∗ | X | | |
| FirstOrder | | | | | X | |
| SecondOrder | | | | | | X |

arises for tests on the *RR* metric. In image collection original, BlockMatch achieves a perfect return ratio of one for all snapshot positions. Thus, the standard deviation is zero and statistical tests are inappropriate. For this reason, all comparisons with BlockMatch are marked ‘NA’ in table 1.

Tables 2 and 3 show that BlockMatch performs significantly better (lower *AAE* and higher *RR*) for both performance metrics on the two hall image collections. Further, table 4 shows that it also performs better than Warping for the cross-collection tests.

From all of the above, it seems safe to conclude that BlockMatch outperforms Warping on our image database.

## 4.9 *Robot Homing Trials*

A series of online homing trials was carried out in order to test the real-world applicability of our results on BlockMatch.

Table 3. Statistical significance of the difference in $AAE_*$ (hall2) and $RR_*$(hall2) between homing methods; see table 1 for notation.

| | Warping | BlockMatch | IntMatch | GradMatch | FirstOrder | SecondOrder |
|---|---|---|---|---|---|---|
| *Significance of difference in* $AAE_*$*(hall2)* | | | | | | |
| Warping | X | | ∗∗∗ | | | |
| BlockMatch | ∗∗∗ | X | ∗∗∗ | ∗∗∗ | ∗ | |
| IntMatch | | | X | | | |
| GradMatch | | | ∗∗∗ | X | | |
| FirstOrder | | | ∗∗∗ | | X | |
| SecondOrder | | | ∗∗∗ | | | X |
| *Significance of difference in* $RR_*$*(hall2)* | | | | | | |
| Warping | X | | | | | |
| BlockMatch | ∗∗∗ | X | ∗∗∗ | | | |
| IntMatch | | | X | | | |
| GradMatch | | | ∗∗∗ | X | | |
| FirstOrder | | | ∗∗∗ | | X | |
| SecondOrder | | | ∗∗∗ | | | X |

Table 4. Statistical significance of the difference in $AAE_*$ $(*, *)$ and $RR_*(*, *)$ between homing methods for all pairings of image collections from the office environment; see table 1 for notation.

| | Warping | BlockMatch | IntMatch | GradMatch | FirstOrder | SecondOrder |
|---|---|---|---|---|---|---|
| *Significance of difference in* $AAE_*(*, *)$ | | | | | | |
| Warping | X | | | | | |
| BlockMatch | $***$ | X | $***$ | | $**$ | $***$ |
| IntMatch | $***$ | | X | | | |
| GradMatch | $***$ | | | X | | |
| FirstOrder | $***$ | | | | X | |
| SecondOrder | $***$ | | | | | X |
| *Significance of difference in* $RR_*(*, *)$ | | | | | | |
| Warping | X | | | | | |
| BlockMatch | $***$ | X | $***$ | | | |
| IntMatch | | | X | | | |
| GradMatch | $*$ | | | X | | |
| FirstOrder | $**$ | | | | X | |
| SecondOrder | $***$ | | $**$ | | | X |

Figure 13 shows traces of the robot homing according to Warping and BlockMatch.[†] The number of successful homing attempts for Warping was 20 out of 24. In the left two plots of figure 13(a) we see that Warping was able to home to the top two snapshot positions from all start positions. However, the right two plots show that homing to the bottom two snapshot positions was successful only from the bottom four start positions. The number of successful homing attempts for BlockMatch was 24 out of 24. While no definitive statements can be made from these data alone, these results do lend credence to the results obtained on the image database. That is, the superior performance of BlockMatch does not appear to be an artefact of the offline testing methodology.

### 4.10 *Analysis*

It was an unexpected result that BlockMatch should perform so well. The vector mapping stage assumes that correct correspondences have been established, and it is surprising that block matching should be able to deliver sufficiently correct correspondences, especially since the image features to be corresponded have undergone significant distortions due to scaling, perspective, occlusion and illumination changes. Computer vision researchers have invented a multitude of image matching strategies dedicated to robustly handling one or all of these distortions (see review by Lowe (2004)). All of this activity would not have taken place if block matching – an easily understood and well-known technique – had been able to satisfy the needs of these researchers. Yet it does seem to satisfy the needs for visual homing, at least within the framework described here.

The approach we now take is to try and understand BlockMatch's success with the goal of replacing it by even simpler methods.

**4.10.1 The democracy effect.** An examination of the correspondence vectors generated by BlockMatch reveals that not all image features correspond correctly. Figure 14 shows a snapshot and current image pair, along with the correspondence vectors computed by BlockMatch

[†] These robot homing trials were carried out on an older variant of BlockMatch which employed a different (generally less successful) configuration.

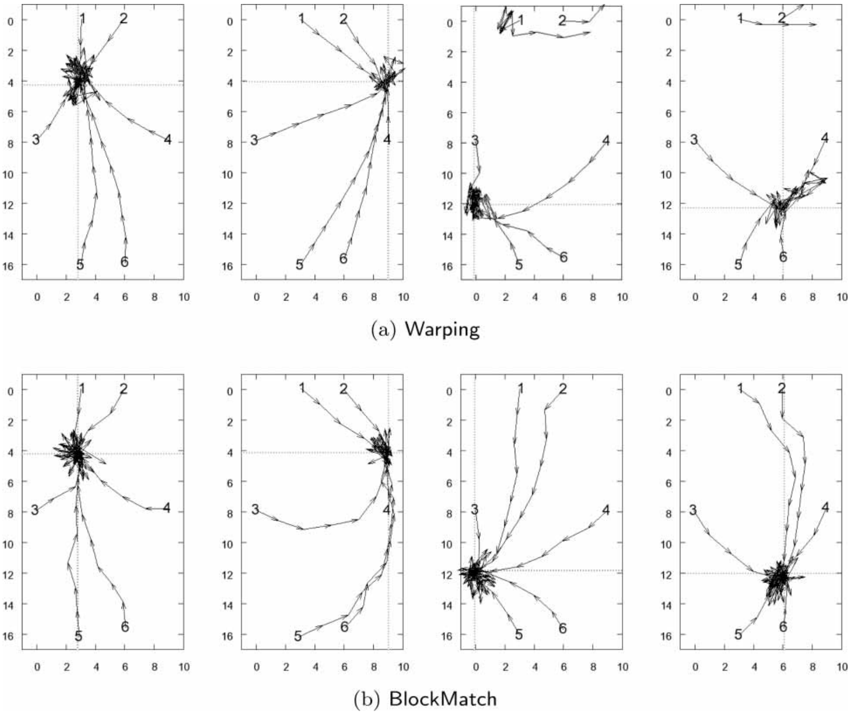


Figure 13. Performance of Warping and BlockMatch when tested live on the Pioneer robot. Numbers indicate start positions and intersecting dashed lines indicate snapshot positions.

between them. The current image was taken from a position to the left of the snapshot position. The image position in the current view corresponding to the correct home direction is (52,23). Thus, at this position we expect to find the *focus of contraction* (FOC) of the flow field. That is, image features will have been drawn in towards the FOC by the displacement of the agent.

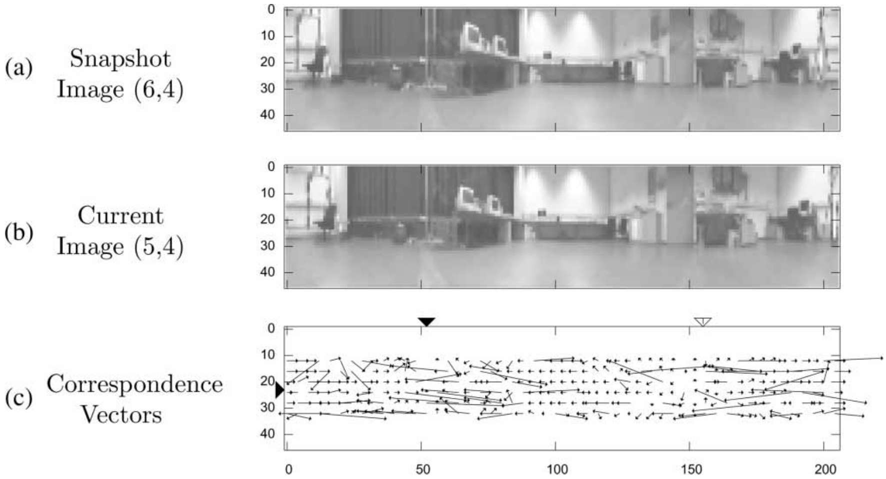


Figure 14. Example flow field for snapshot position (6,4) and current position (5,4). The correct home vector is directed towards the right, meaning that the correct FOC should be around (52,23) while the correct FOE should be near (155,23). These positions are indicated in (c) by triangles along the plot boundary. The horizontal position of the FOC is indicated by the closed triangle, while the FOE is indicated by the open triangle.

Likewise, at (155,23) we expect to see the *focus of expansion* (FOE). Indeed, contraction can be observed around the FOC and expansion around the FOE. However, there are many clearly incorrect vectors. The situation gets worse as the distance to the goal increases. Figure 15 shows flow fields for positions further and further to the left of the goal. However, as can be seen in figure 7(b), the computed home vector is still appropriately directed towards home for these same positions.

The method is able to cope with incorrect correspondences as long as the overall number of correspondence vectors remains high. To test this hypothesis, the step size, $m = m_x = m_y$, was varied in increasing powers of two. We increased step size in powers of two so that each set of correspondence vectors was a subset of the previous set of vectors – this would not be the case for a linear increase in step size. The total number of origin points decreases quadratically with increasing step size.

Figure 16(a) shows a clear increase in $AAE_*$(original) for increasing step size. This effect stems from the uncorrelated mismatches assumption, described in section 4.3. As step size increases, the number of origin points decreases. With a small number of origin points, the incorrect correspondences will cancel each other out less effectively. Further, there may be a change in the ratio of correct to incorrect correspondences if we sample the image less densely. Thus, as we include more origin points, each 'voting' for home vectors, the final summed home vector becomes more correct. We refer to this as the *democracy effect*.

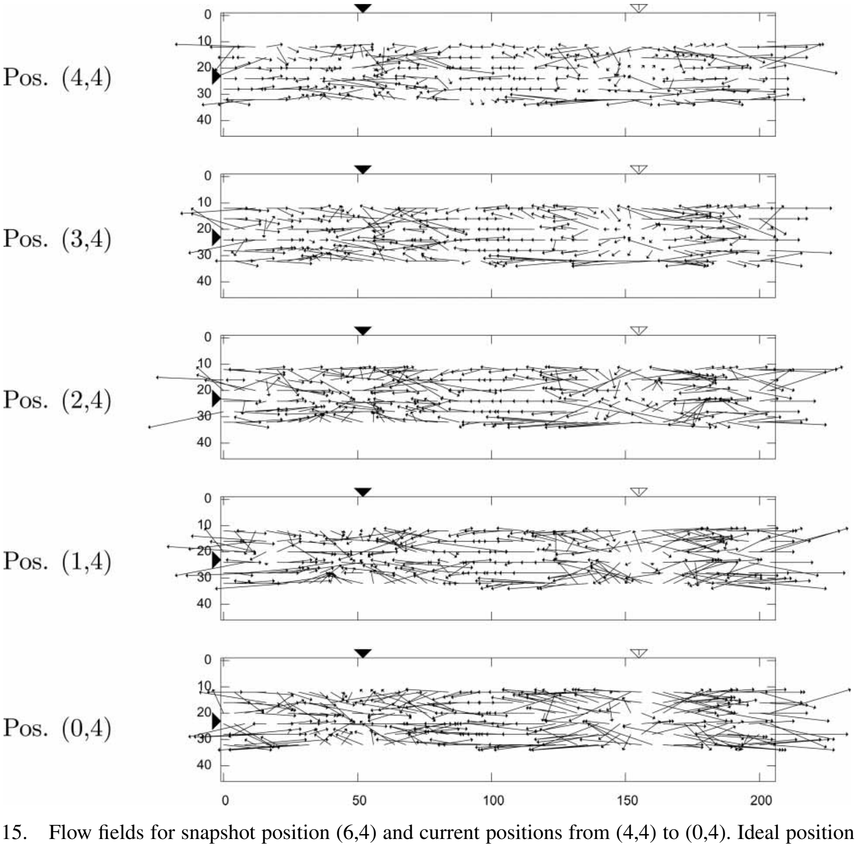
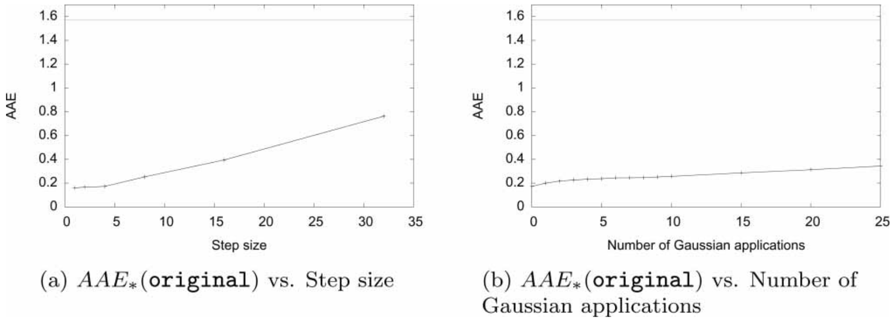


Figure 15. Flow fields for snapshot position (6,4) and current positions from (4,4) to (0,4). Ideal positions of the FOC and FOE are indicated as in figure 14.



(a) $AAE_*(\texttt{original})$ vs. Step size

(b) $AAE_*(\texttt{original})$ vs. Number of Gaussian applications

Figure 16. Plots of angular error versus parameters. A dashed line is shown at $\pi/2$.

**4.10.2 Frequency content.** It was initially assumed that it was the high-frequency structure of image patches that allowed blocks to be distinguished from each other and matched correctly. However, this turns out not to be the case. If the high-frequency structure of local image patches was indeed the basis for BlockMatch's success, then we should see performance decline as this high-frequency structure is filtered out. The following experiment reveals only a slight decline in performance when the high-frequency structure is filtered out.

BlockMatch was tested with parameters as described previously, with the exception that the input is filtered by repeated application of a Gaussian filter. Repeated convolution of a signal by a Gaussian kernel is equivalent to convolution of that signal with a wider Gaussian kernel. So by repeatedly applying a Gaussian filter, more and more of the high-frequency content is filtered out of the signal. This Gaussian filter uses the same kernel described in the footnote for section 4.4, but at the top and bottom image boundaries, out-of-bounds pixels are treated as zeros. This introduces some boundary effects (visible in figure 17) but prevents the image from shrinking vertically, therefore allowing the filter to be applied many times.

Figure 16(b) shows the value of $AAE_*$(original) for an increasing number of filter applications. The angular error increases somewhat with more filtering, but even with very heavy filtering it remains far below $\pi/2$. To give a visual impression, figure 17 shows image (6,4) from image collection original and the effects of five and 25 filtering operations on this image.

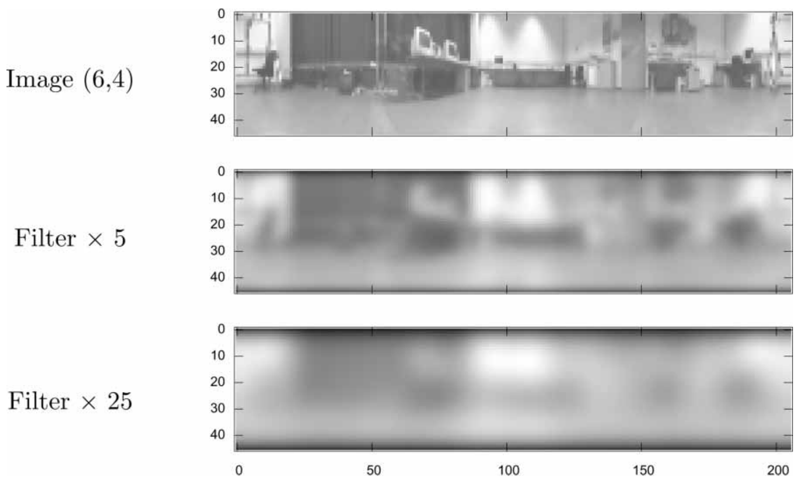


Figure 17. The effects of increased Gaussian filtering on image (6,4) from image collection original.

The experiment shows that it must be the low-frequency image components that BlockMatch matches between images. By definition, low-frequency components are those that vary slowly across the image.

### 4.11 *Complexity and biological plausibility*

**4.11.1 BlockMatch.** While BlockMatch is conceptually simple, it requires an expensive search for each correspondence. On a PC with a Pentium 4 2 GHz processor it takes approximately 327 ms to compute a home vector.[†] In terms of pixel-to-pixel comparisons, each correspondence search requires a number of comparisons equal to

$$\mathcal{C}_{\text{search}} = (2r + 1)^2(2q + 1)^2. \tag{17}$$

The total number of operations required to compute a home vector is

$$\mathcal{C}_{\text{BlockMatch}} = \mathcal{C}_{\text{search}} n_x n_y. \tag{18}$$

For the parameters selected in section 4.5, the quantity above comes to almost 20 million. We find it implausible that this sort of algorithm could be implemented in the limited neural hardware of an insect. Worker honeybees have on the order of 850 000 neurons (Hertel 1987). If we assume that one neuron can compute one pixel to-pixel comparison, it remains clearly impossible for all operations to be carried out in parallel. A serialized execution is possible in theory but contradicts the generally held view of parallel processing in animal brains.

**4.11.2 Warping.** The fact that Warping operates on one-dimensional images makes it tractable for realtime operation. The averaged processing time per image on a Pentium 4 2 GHz PC was 16.8 ms.

The number of pixel-to-pixel comparisons required for Warping is:

$$\mathcal{C}_{\text{Warping}} = n_\alpha n_\rho n_\psi \lfloor w/m_x \rfloor. \tag{19}$$

For the parameters selected in section 4.6 this quantity comes to $64^2 \times 206 \approx 850\,000$ operations. Franz *et al.* (1998b) argue that their method could conceivably be implemented in parallel through a set of *matched filters*, akin to those described for optic flow determination in insects (Krapp and Hengstenberg 1996, Krapp 2000, Franz and Krapp 2000). However, 850 000 operations would still be required, and if they were all to occur in parallel using at least one neuron per operation then Warping appears rather infeasible for implementation in the brain of an insect.

## 5. Homing by intensity and gradient matching

### 5.1 *Computation of correspondence vectors*

We conclude from the experiment of section 4.10.2 that BlockMatch actually pairs blocks whose low-frequency components match. The simplest low-frequency component of a block is just its overall intensity. The next simplest low-frequency component is the change in

[†] The same computer was used for all processing times reported in this paper. Estimates of average processing time were obtained by processing all images from image collection original and dividing by the number of images in this collection (170).

overall intensity, which is a two-dimensional quantity given by the gradient of the intensity. We implemented two new homing methods based on matching the intensity and matching the gradient of intensity. We shall refer to these new methods as 'IntMatch' and 'GradMatch', respectively. IntMatch is essentially the same as BlockMatch with a block radius of zero. To assess the quality of a potential match between **p** and **p**′ we do not compute the SSD but instead compute

$$|S(\mathbf{p}) - C(\mathbf{p}')|. \tag{20}$$

GradMatch involves the additional step of computing the gradient of the image intensity. The image gradient is defined as the vector of first derivatives of the image in the $x$ and $y$ directions. For an image $I$, the gradient is

$$\nabla I = \left( \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right)^{\mathrm{T}}. \tag{21}$$

The quantity $\partial I/\partial x$ is estimated by convolving $I$ with the kernel $(1\,0\,-1)$. Similarly, $\partial I/\partial y$ is estimated by convolution with $(1\,0\,-1)^T$. Convolution by these kernels implements the central-difference approximation of the first derivative (see Appendix A.2 of Trucco and Verri 1998).

For GradMatch the gradient is computed for both the snapshot and current images. The quality of a match is judged by the quantity

$$\|\nabla S(\mathbf{p}) - \nabla C(\mathbf{p}')\|. \tag{22}$$

While the entities matched between snapshot and current image differ for BlockMatch, IntMatch and GradMatch, they all find correspondence vectors between images and then apply the same vector mapping and summation stages to arrive at a final home vector.

## 5.2 *Optimizing parameters*

The parameters tested for IntMatch and GradMatch are the same as for BlockMatch:

- Number of applications of Gaussian filter: **0**, 1, 2, <u>3</u>, or 4.
- Use Sobel filter: true or <u>**false**</u>.
- Number of top and bottom 10 image rows to exclude: <u>**0**</u>, 5, or 10.
- Search radius, $q$: 5, **10**, <u>30</u>, or 50.

The best parameters found for IntMatch are given in bold. The best parameters for GradMatch have been underlined. Other parameters, such as step size, were set as for BlockMatch.

## 5.3 *Results on image database*

### 5.3.1 Image collection original.

Both IntMatch and GradMatch perform remarkably well on image collection original. Figure 8 plots *AAE* and *RR* for these methods across the 20 snapshot positions. These plots confirm the theory that matching these very simplistic image descriptors can suffice for visual homing. The plots for IntMatch in particular show no areas were performance is weak. The plots for GradMatch show a slight degradation in performance for the bottom portion of the capture grid.

**5.3.2 Image collections hall1 and hall2.** Figures 10 and 11 plot our performance metrics across snapshot positions for the two hall image collections. GradMatch stands out in these plots as exhibiting low *AAE* and high *RR* for a majority of snapshot positions. IntMatch does not appear to perform as strongly, particularly in the top portion of hall2. It is clear that neither method performs as well as BlockMatch.

**5.3.3 Cross-collection tests.** When snapshot and current images are taken from different image collections, the two simplified matching methods still perform quite impressively. Figure 12 presents the relevant plots. With some variation, both IntMatch and GradMatch perform in a qualitatively similar fashion to BlockMatch.

### 5.4 *Comparative results*

Table 1 presents the results of statistical analyses for all methods considered in this paper on image collection original. For both performance metrics, IntMatch and GradMatch perform significantly better than Warping. For the *AAE* metric, no significant difference is found with BlockMatch.

Table 2 presents the results of the same tests for image collection hall1. In this case, IntMatch has a significantly *higher AAE* than Warping, BlockMatch and GradMatch. However, for the *RR* metric the difference between IntMatch and Warping is no longer significant. The results are the same for image collection hall2 (see table 3) with the exception that BlockMatch has a significantly lower *AAE* than GradMatch.

Table 4 presents statistical analyses of the cross-collection tests. In terms of *AAE* both IntMatch and GradMatch outperform Warping. However, for the *RR* metric the difference between Warping and IntMatch is not significant. BlockMatch performs significantly better than IntMatch for both performance metrics, but the differences in performance with GradMatch are not significant.

In all of these tests, BlockMatch remains unbeaten. GradMatch also does quite well and is only ever outperformed by BlockMatch. IntMatch is outperformed by these two methods in various cases. However, it is outperformed by Warping only in the hall collections and only for the *AAE* metric.

While neither IntMatch nor GradMatch outperform BlockMatch, they have both been shown to perform to a high standard. As GradMatch is never beaten by Warping it joins BlockMatch as a method that is competitive with Warping. The case for IntMatch is less clear-cut but it remains a very interesting method.

It may be that BlockMatch should be viewed as a hybrid of IntMatch, GradMatch and perhaps even high-frequency matching – the result in section 4.10.2 showed that high frequencies were not the primary component of block matching's success; but they may well be useful for certain kinds of images. Thus, in certain environments BlockMatch would employ intensity matching, while in others gradient matching or high-frequency matching might be preferred.

### 5.5 *Complexity and biological plausibility*

The computational complexities for IntMatch and GradMatch are clearly reduced from BlockMatch. These methods take an average of 86.5 and 224.5 ms, respectively, to compute a home vector on the same computer as used above.

Where $f = 1$ for IntMatch and $f = 2$ for GradMatch, the number of operations required to compute a home vector is

$$\mathcal{C}_{\text{IntMatch,GradMatch}} = f(2q+1)^2 n_x n_y. \tag{23}$$

This does not include the cost of pre-processing. For the parameters selected for IntMatch, this quantity comes to approximately 260 000. For GradMatch the cost is about 4.4 million. The search radius selected for IntMatch was 10, as opposed to 30 for GradMatch. The cost of BlockMatch with a search radius of 10 is approximately 520 000.

Clearly these methods are more biologically plausible for insects than BlockMatch. However, it still seems quite unlikely that 260 000 of the 850 000 neurons in a honeybee's brain would be dedicated to this one task of visual homing. Indeed, if this were the case then the neural apparatus implementing the correspondence search process would be quite prevalent in neurophysiological studies. While such studies do reveal extensive horizontal interconnections between columns of the retinotopic lattice that characterizes the optic lobes of bee brains (Hertel 1987, Ribi 1987), it seems unlikely that the dense interconnectivity required for correspondence search would have gone unnoticed.

## 6. Homing by differential methods

Encouraged by the performance of IntMatch and GradMatch, we investigated whether their differential counterparts would also produce reliable visual homing. Differential techniques assume that image brightness is conserved between the snapshot and current images. They also assume that all feature displacements are relatively small. Despite these two strong assumptions, it turns out that differential techniques are very useful for visual homing. The advantages of these techniques are that they do not require a search process and that all computations are purely local.

### 6.1 *Computation of correspondence vectors*

The method based on IntMatch employs first-order derivatives to estimate the direction of feature translation. Thus, we refer to this method as 'FirstOrder'. The method based on GradMatch employs second-order derivatives and is therefore referred to as 'SecondOrder'. For FirstOrder, the feature that experiences translation is just a pixel with a particular intensity. For SecondOrder, the feature that experiences translation is a vector – the gradient of intensity. Finding the actual translation of a feature from the snapshot image to the current image is equivalent to finding a correspondence vector. Appendix A presents these two differential methods and shows how they compute correspondence vectors.

The length of correspondence vectors estimated by the differential methods was thought to be relatively inaccurate. The exact vector mapping technique presented in section 4.2 requires the length of correspondence vectors to be known. Therefore, a new approximate vector mapping technique was developed. This new technique is presented in section B.2 in the Appendix.

### 6.2 *Assumptions*

For FirstOrder, it is assumed that the pixel intensity is conserved between the snapshot and current images. For SecondOrder, it is assumed that the gradient of intensity is conserved (which implicitly assumes that intensity is also conserved). These assumptions are bound to be unfulfilled in many real situations. However, it is one of the basic assumptions of visual homing that *something* is conserved between images.

More troubling, perhaps, is the assumption of small feature translations. Both methods employ a Taylor series approximation that is valid only for small shifts of the intensity, or gradient of intensity, between images. This assumption is clearly not fulfilled if the two images are taken from disparate positions in the environment. Does this constrain the differential methods to be effective only in the immediate vicinity of the goal? Perhaps not, as there are two factors that justify this assumption.

The first factor becomes apparent when we consider the formula for the error of a Taylor approximation. The error for a first-order Taylor approximation is defined as follows (Stewart 1991):

$$\left|\frac{1}{2}f''(z)(x-c)^2\right|, \tag{24}$$

where $f$ is the function approximated and the goal is to approximate $f(x)$. The value of $f$ and its first derivative are known at $c$. The quantity $z$ is an unknown bound only to lie somewhere in the range $[x, c]$. The term $(x - c)$ is the shift between the position where $f$ is known $(c)$, and where the approximation is desired $(x)$. The fact that error is proportional to $(x - c)^2$ implies that the size of shift must be kept small. However, for any given shift the error will depend on the size of $f''(z)$. A function with small derivatives will therefore exhibit lower error. By definition, a low-frequency function has small derivatives. This analysis holds if we consider not a one-dimensional function but a two-dimensional image.

Thus, errors in the Taylor approximation are reduced for images with significant energy in the low-frequency band. It has already been demonstrated that sufficient low-frequency components exist in images from the Bielefeld database for methods like IntMatch and GradMatch to succeed. Further, it is known that natural images exhibit a strong portion of low frequencies (Ruderman and Bialek 1994, van der Schaaf and van Hateren 1996). This gives us reason to believe that the differential methods will be successful.

The second factor relates to the geometry of visual homing. Nelson and Aloimonos (1988) described the optic flow experienced by a moving agent with a spherical field of view. In the case of pure translation, image features will move along great circles joining the two foci of the visual field. These two foci are the *focus of expansion* (FOE) and the *focus of contraction* (FOC). Image features flow away from the FOE and toward the FOC. If there has been no rotation then the FOE and FOC will be 180° apart. Image regions near the FOE and FOC will exhibit relatively little movement. Thus, for image regions near the foci, the requirement of small shifts will be satisfied. These small shifts will ideally generate approximately correct correspondence vectors. From these approximately correct correspondence vectors we shall obtain approximately correct home vectors all directed in a consistent direction. Image regions with large shifts away from the foci will ideally generate uncorrelated correspondence vectors and therefore uncorrelated home vectors. Thus, the uncorrelated home vectors will cancel themselves out and leave an approximately correct home vector. This is the assumption of uncorrelated mismatches from section 4.3.

### 6.3 *Optimizing parameters*

The parameters tested for both FirstOrder and SecondOrder are as follows:

- Number of applications of Gaussian filter: 0, 1, 2, 3, **4**, or 5.
- Number of top and bottom 10 image rows to exclude: **0**, 5, or 10.

The best parameters found for FirstOrder are given in bold. The best parameters for SecondOrder have been underlined. As these methods are computationally cheap, we opt to compute correspondence vectors at all image positions, as opposed to a subset of image positions as was done for the matching methods. This again is intended to take advantage of the democracy effect.

## 6.4 *Results on image database*

**6.4.1 Image collection original.** Figure 8 shows the performance of FirstOrder and SecondOrder across snapshot positions of image collection original. FirstOrder performs quite well across all positions. SecondOrder performs less well, but still exhibits relatively low *AAE* and high *RR* across most positions.

**6.4.2 Image collections hall1 and hall2.** Figures 10 and 11 show the performance of the differential methods across snapshot positions for the hall image collections. Both methods exhibit small areas where performance is degraded. However, the overall level of performance remains quite high for both methods.

**6.4.3 Cross-collection tests.** Figure 12 plots $AAE_*$ and $RR_*$ for the cross-collection tests. The performance of both differential methods remains impressive for these tests. FirstOrder shows low *AAE* and particularly high *RR* for all pairs of image collections except those involving day, doorlit, or winlit. SecondOrder shows remarkably constant performance in both performance metrics for all pairs of image collections, *including* those involving day, doorlit, and winlit.

## 6.5 *Comparative results*

Table 1 shows that on image collection original, FirstOrder performs significantly better than Warping and SecondOrder for both performance metrics. SecondOrder performs relatively poorly on original and is beaten by most other methods. However, the difference in performance with Warping is not significant for either metric.

For image collection hall1 (see table 2) BlockMatch performed significantly better than both differential methods in *AAE*, but not in *RR*. Also, both FirstOrder and SecondOrder performed significantly better than IntMatch in *AAE*, but not in *RR*. No other significant differences between the differential methods and the other methods were found. The results for image collection hall2 (see table 3) were similar, except that no significant differences were found between BlockMatch and SecondOrder, and that the differential methods outperformed IntMatch for both performance metrics.

Statistical analysis of the cross-collection tests is reported in table 4. For both performance metrics FirstOrder and SecondOrder outperform Warping. Also, for the $RR$ metric, SecondOrder outperforms IntMatch. For the *AAE* metric, BlockMatch performs significantly better than the differential methods.

Both differential methods exhibit extremely impressive performance, despite their simplicity. In no case does Warping exhibit better performance. Indeed, Warping is competitive with the differential methods only in the hall image collections. Otherwise, one or both of the differential methods outperformed Warping. The only significant differences found between FirstOrder and SecondOrder were on image collection original where FirstOrder was found to be superior.

## 6.6 *Analysis*

The results indicate that the underlying assumptions of the differential methods must have been satisfied to some extent. In particular, some interesting insight into the operation of these methods is gained by examining how the angular error changes with respect to the location of the foci. Recall that the requirement for small feature shifts should be most satisfied near the foci.

From the perspective of analysis, we know the true home vector. We can compare this vector with every computed home vector and find the angle between them. As both differential methods compute home vectors at every pixel, we can compute an *error image* for every processed image. Figure 18 shows error images for both FirstOrder and SecondOrder at snapshot position (6,8) and current position (5,8) of image collection original. In figure 18(a) the amount of error is not noticeably different near the foci than for other regions of the error image. In figure 18(b), the error does appear somewhat lower near the foci.

Examining single images such as those in figure 18 is inconclusive. A single error image is specific to a particular pair of snapshot and current positions. We can obtain a more robust indication of where the error lies by combining error images from multiple pairs of snapshot and current positions. Three thousand four hundred such pairs were obtained for image collection original by selecting one of the collection's 170 images as the current image, and an image from one of the 20 positions shown in figure 4 as the snapshot image. Three thousand four hundred error images were generated for both differential methods. As we are interested in where the error lies with respect to the FOC and FOE, each of these error images was rotated such that the ideal location of the FOC was at position zero, and that of the FOE was at position $w/2$, where $w$ is the width of the image. An average error image can then be obtained by summing all 3400 rotated error image and dividing each pixel by 3400.

Figure 19 shows average error images from both differential methods. Also shown are one-dimensional plots from the same data, obtained by averaging the columns of the average error images. It is clear that error is lower at the locations of the FOC and FOE. This lends credence to the assumption of small shifts at the foci. The fact that we are using a panoramic imaging system is important because it means that both foci are always in view. A narrow field of view could dramatically reduce the effectiveness of the differential methods. Yet a complete 360° image may not be required. If the imaging system has just a hemispherical (180°) field of view then one of the foci will always be in view. For a larger field of view such as that of the honeybee (ca. 310°, Land 1989), both foci would be in view for most orientations.

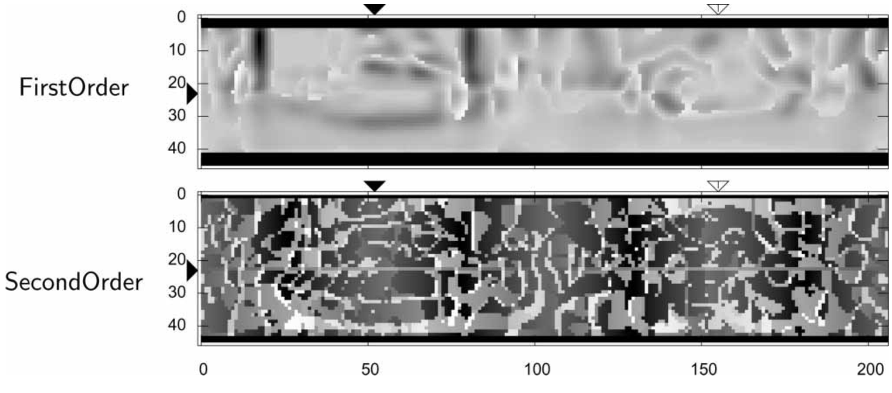


Figure 18. Error images for both differential methods at snapshot position (6,8) and current position (5,8). Lighter pixels indicate higher angular error. Ideal positions of the FOC and FOE are indicated as in figure 14.

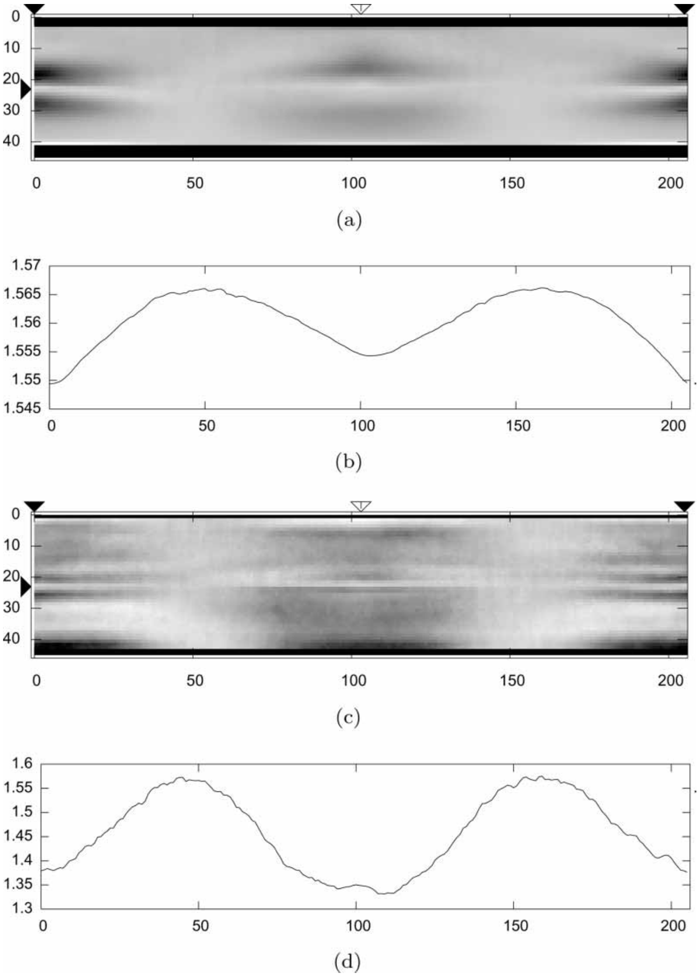


Figure 19. Average error images and one-dimensional plots for both differential methods on image collection original. (a) Average error image for FirstOrder; (b) Sum of columns of image in (a); (c) Average error image for SecondOrder; (d) Sum of columns of image in (c).

## 6.7 *Complexity and biological plausibility*

The differential methods require no search for correspondences. This search was the primary component in previous measures of computational complexity. For the differential methods, the computational cost per pixel depends on the kernels applied and the implementation details of the required functions. A rough analysis of FirstOrder in combination with the approximate vector mapping method reveals that only about 34 low-level operations (*e.g.* additions, multiplications, $\sqrt{\cdot}$, sgn()) are required per pixel. At each pixel an estimated home vector is computed. These are finally summed together to produce the final home vector. This summing process can be incorporated into the per pixel cost, yielding 35 low-level operations for the first-order method. For SecondOrder, about 51 operations are required.

On a Pentium 4 2 GHz PC, FirstOrder requires 193.7 ms on average to process an image, while SecondOrder requires 302.1 ms. If the rather expensive Gaussian filter is removed then these processing times drop to 100.5 and 165.5 ms, respectively. While these processing times should be sufficiently low for real-time operation on a mobile robot, it is notable that the time required by Warping (16.8 ms) is still much lower. Warping operates on one-dimensional images which has a profound effect on processing time.

The visual systems of insects are arranged in a retinotopic manner, with repeated and identical processing columns lying beneath each facet of the compound eye (Strausfeld 1976, Hertel 1987, Ribi 1987). Algorithms like the differential methods whose per pixel cost is low would map well on to this structure. On the other hand, methods with a high per pixel cost such as the matching methods would not. For Warping the mapping to a per pixel cost is not as natural. The per pixel cost for each pixel of a one-dimensional image equals the density of the search space ($64^2$, in our case).

## 7. Discussion

All optical flow techniques studied in this paper establish correspondences between features in two images. From the perspective of visual homing, they are therefore related to the original snapshot model. With respect to the selection of features, our optical flow techniques take middle ground. On the one hand, they do not focus on distinctive features in the matching process; rather, features at *all* positions in the image are involved in the matching. In this respect, they are simpler than correspondence approaches coming from computer vision. On the other hand, optical flow methods do not restrict themselves to a few classes of features. Thus, they are more complex than the snapshot model which only distinguishes between dark and bright regions. Projecting all features on to a small number of classes allows simplification of the matching process to a search for feature positions, which can be performed independently for each class. In contrast, if features are just continuous vectors, as in optical flow techniques, matching requires a costly minimum search over some distance measure between vectors. However, our analysis of BlockMatch showed that a matching between low-frequency image features is sufficient for homing. This was corroborated by the success of methods IntMatch and GradMatch. Thus, the complexity of the matching process can be reduced by just considering one value (intensity) or two values (gradient) for the correspondence measure. While the performance results on IntMatch vary over the three image collections, the performance of GradMatch is significantly different from that of BlockMatch only for image collection hall2, and only for the *AAE* metric.

The reduction of block matching to intensity matching and gradient matching turned our attention towards differential optical flow methods. As shown in Appendix A, IntMatch has its counterpart in FirstOrder, while GradMatch corresponds to SecondOrder. The convergence of several factors allows the differential methods to succeed. The prevalence of low-frequency image features reduces the error due to the Taylor approximation. This error is reduced further wherever the shift of features between the snapshot and current images is small, which occurs near the two foci of the translatory flow field. The use of panoramic images guarantees that both foci are always in view. Finally, the uncorrelated mismatches assumption permits successful homing despite deviations from these assumptions.

The finding that differential-based techniques can be used for visual homing comes as quite a surprise. Such techniques have been in use by the computer vision community since at least 1980 (Horn and Schunck 1980). They have been employed in the study of ego-motion (Gluckman and Nayar 1998) and in inferring the structure of the environment from the optic flow field (see Trucco and Verri 1998: chapter 8). However, it was always assumed that accurate flow fields can be estimated by differential flow fields only for small movements. Also, these methods exhibit the *aperture problem* whereby flow can be estimated only in the direction of the image gradient. Therefore, differential techniques were naturally excluded from consideration for visual homing. At the time of writing, no record of their use in visual homing could be found.

What makes differential techniques particularly attractive is their plausibility as models of insect navigation. So far, nothing is known about the neural apparatus underlying visual landmark navigation in insects, but block matching in the complexity range tested here can be ruled out as a model of insect navigation just on the basis of the number of required neurons, which exceeds by far the size of insect brains. When interpreted as neural architecture, differential techniques are collections of functionally equivalent local networks, each receiving visual input from small receptive fields in both the stored and the current image and producing as output the correspondence vector or the individual home vector in some suitable representation. It is conceivable that such an architecture could be projected on to the crystalline, repetitive, retinotopic structure of the visual ganglia of insect brains (Strausfeld 1976, Hertel 1987, Ribi 1987).

From the application perspective, our implementation of block matching deserves attention. Throughout all image collections, no method was significantly better than BlockMatch which, in turn, was significantly better than Warping in all cases. The excellent performance of this method has been verified through online robotic trials. GradMatch, FirstOrder and SecondOrder also perform quite well – exhibiting performance as good or better than Warping for all image collections.

Although we did not directly compare our optical flow techniques on two-dimensional images with similar methods applied to one-dimensional panoramic views (gained from averaging over columns of two-dimensional images), we suspect that one-dimensional methods (like the method proposed by Hong *et al.* (1992)) will perform worse. This guess is based on our result that performance increases with increasing number of correspondence vectors. Whether a pre-selection of distinctive features, as in Hong *et al.*'s method, compensates for this would have to be analysed. Moreover, two-dimensional images allow the implicit incorporation of the distance to landmarks by measuring the elevation of the corresponding features in both images, as in our exact vector mapping method.

A drawback of the optical flow methods studied in this paper is the necessity to align the two images with respect to an external compass reference. While this is less problematic when looking at the methods as models of insect navigation – insects do have a compass sense – it leads to performance losses for indoor navigation where magnetic compasses are known to be unreliable. However, the results of Zeil *et al.* (2003) demonstrate that a compass estimate can also be obtained from a simple correlation of images. Some exploratory studies show that this 'visual compass' may also produce useful results on our database; a systematic analysis of the visual compass in conjunction with our methods remains to be done. Note that, although the Warping method does not need a compass reference, the version used here did not search through orientation changes and thus also relies on aligned views.

The differential methods in particular will receive our attention in further investigations. Improved performance could be obtained by combining first- and second-order methods (Barron *et al.* 1994). Moreover, similar to the work of Hafner (2001, 2004), the local processing units could be parameterized and optimized with respect to overall homing performance by approaches such as neural network training or evolutionary strategies.

## 8. Conclusion

Simple optical flow techniques – block, intensity and gradient matching, along with first- and second-order differential techniques – produce reliable visual homing behaviour. For the matching methods, the simplicity and lack of distinctiveness of matched features appear to be overcome by the uncorrelated mismatches assumption. Surprisingly simple low-frequency 'features' such as intensity and intensity gradient sufficed for successful homing. Block matching, gradient matching and the differential methods have all turned out to be more reliable than Franz *et al.*'s Warping method and are therefore recommendable as technical solutions to visual homing. Moreover, the differential methods produce correspondence vectors without search; this makes them attractive as models of insect navigation and opens up new and simpler approaches to visual homing in robots.

## Acknowledgements

We are very grateful to Sonja Bisch-Knaden for the statistical analysis of our data. Many thanks also to Frank Röben, Wolfram Schenck and Tim Köhler for discussions and technical support. This work has been supported by scholarships from NSERC (PGS-B 232621 - 2002) and the DAAD (A/04/13058).

## References

A. Anderson, "A model for landmark learning in the honey-bee", *J. Comp. Physiol. A*, 114, pp. 335–355, 1977.

J. Barron, D. Fleet and S. Beauchemin, "Performance of optical flow techniques", *Int. J. Comput. Vision*, 12, pp. 43–77, 1994.

J. Borenstein, H. Everett and L. Feng, '*Where am I? Sensors and methods for mobile robot positioning*', Technical Report, Michigan: University of Michigan, 1996.

B. Cartwright and T. Collett, "Landmark learning in bees", *J. Comp. Physiol. A*, 151, pp. 521–543, 1983.

B. Cartwright and T. Collett, "Landmark maps for honeybees", *Biol. Cybernet.*, 57, pp. 85–93, 1987.

R. Cassinis, D. Grana and A. Rizzi, "Using colour information in an omnidirectional perception system for autonomous robot localization", in *Proceedings of EUROBOT'96*, IEEE Computer Society Press, 1996, pp. 172–176.

C. Colios and P. Trahanias, "A framework for visual landmark identification based on projective and point-mutation invariant vectors", *Robot. Auton. Syst.*, 35, pp. 37–51, 2001.

T. Collett, "Insect navigation *en route* to the goal: multiple strategies for the use of landmarks", *J. Exp. Biol.*, 199, pp. 227–235, 1996.

T. Collett and J. Baron, "Biological compasses and the coordinate frame of landmark memories in honeybees", *Nature*, 368, pp. 137–140, 1994.

T. Collett and M. Collett, "Memory use in insect visual navigation", *Nature Rev. Neurosci.*, 3, pp. 542–552, 2002.

H. Cruse, "A recurrent network for landmark-based navigation", *Biol. Cybernet.*, 88, pp. 425–437, 2003.

M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, 24, pp. 381–395, 1981.

M. Franz and H. Krapp, "Wide-field, motion-sensitive neurons and matched filters for optic flow fields", *Biol. Cybernet.*, 83, pp. 185–197, 2000.

M. Franz and H. Mallot, "Biomimetic robot navigation", *Robot. Auton. Syst.*, 30, pp. 133–153, 2000.

M. Franz, B. Schölkopf, H. Mallot and H. Bülthoff, "Learning view graphs for robot navigation", *Auton. Robots*, 5, pp. 111–125, 1998a.

M. Franz, B. Schölkopf, H. Mallot and H. Bülthoff, "Where did I take that snapshot? Scene-based homing by image matching", *Biol. Cybernet.*, 79, pp. 191–202, 1998b.

P. Gaussier, C. Joulain, J. Banquet and A.S.L. Revel, "The visual homing problem: an example of robotics/biology cross fertilization", *Robot. Auton. Syst.*, 30, pp. 155–180, 2000.

J. Gluckman and S. Nayar, "Ego-motion and omnidirectional cameras", in *Proceedings of the International Conference on Computer Vision*, 1998, pp. 999–1005.

T. Goedemé, M. Nuttin, T. Tuytelaars and L. Van Gool, "Vision-based intelligent wheel chair control: rhe role of vision and inertial sensing in topological navigation", *J. Robot. Syst.*, 21, pp. 85–94, 2004.

S. Gourichon, J.-A. Meyer, S. Ieng, L. Smadja and R. Benosman, "Estimating ego-motion using a panoramic sensor: comparison between a bio-inspired and a camera-calibrated method", in *AISB03 Symposium on Biologically Inspired Vision, Theory and Application*, 2003, pp. 91–101.

S. Gourichon, J.-A. Meyer and P. Pirim, "Using colored snapshots for shortrange guidance in mobile robots", *Int. J. Robot. Automation*, 17, pp. 154–162, 2002.

V. Hafner, "Adaptive homing – robotic exploration tours", *Adaptive Behav.*, 9, pp. 131–141, 2001.

V. Hafner, "Adaptive navigation strategies in biorobotics: visual homing and cognitive mapping in animals and machines", PhD thesis, University of Zurich (2004).

R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge: Cambridge University Press, 2004.

H. Hertel, "Processing of visual information in the honeybee brain", in *Neurobiology and Behavior of Honeybees*, R. Menzel and A. Mercer, Eds, Berlin: Springer, 1987, pp. 141–157.

J. Hong, X. Tan, B. Pinette, R. Weiss and E. Riseman, "Image-based homing", *IEEE Control Syst.*, 12(1), pp. 38–45, 1992.

B. Horn and B. Schunck, "Determining optical flow", Technical Report 572, Artificial Intelligence Laboratory, MIT (1980).

J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, 29, pp. 1799–1808, 1981.

M. Jogan and A. Leonardis, "Panoramic eigenimages for localisation", in *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science 1689*, Berlin: Springer, 1999, pp. 558–567.

D. Kortenkamp, "Cognitive maps for mobile robots: a representation for mapping and navigation", PhD thesis, University of Michigan (1993).

H. Krapp, Neuronal matched filters for optic flow processing in flying insects", *Int. Rev. Neurobiol.*, 44, pp. 93–120, 2000.

H. Krapp and R. Hengstenberg, "Estimation of self-motion by optic flow processing in single visual interneurons", *Nature*, 384, pp. 463–466, 1996.

B. Kuipers and Y.-T. Byun, "A robust, qualitative approach to a spatial learning mobile robot", *Sens. Fusion: Spatial Reasoning Scene Interp. Proc SPIE*, 1003, pp. 366–375, 1988.

D. Lambrinos, "Navigation in biorobotic agents" PhD thesis, University of Zurich (1999).

D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer and R. Wehner, "A mobile robot employing insect strategies for navigation", *Robot. Auton. Syst.*, 30, pp. 39–64, 2000.

M. Land, "Variations in the structure and design of compound eyes", in *Facets of Vision*, D.G. Stavenga and R.C. Hardie, Eds, Berlin: Springer, 1989, Chapter 5.

M. Land, "Visual acuity in insects", *Ann. Rev. Entomol.*, 42, pp. 147–177, 1997.

M. Lehrer and G. Bianco, "The turn-back-and-look behaviour: bee versus robot", *Biol. Cybernet.*, 83, pp. 211–229, 2000.

D. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vision*, 60, pp. 91–110, 2004.

R. Möller, "Insect visual homing strategies in a robot with analog processing", *Biol. Cybernet.*, 83, pp. 231–243, 2000.

R. Möller, "Do insects use templates or parameters for landmark navigation?", *J. Theor. Biol.*, 210, pp. 33–45, 2001.

R. Möller, "A biorobotics approach to the study of insect visual homing strategies", Habilitationsschrift, Universität Zürich (2002a).

R. Möller, "Insects could exploit UV-green contrast for landmark navigation", *J. Theor. Biol.*, 214, pp. 619–631, 2002b.

R. Möller, "Visual homing without image matching", in *Neurotechnology for Biomimetic Robots*, J. Ayers, J.L. Davis and A. Rudolph, Eds, Cambridge, MA: MIT Press, 2002c, pp. 517–532.

R. Möller, D. Lambrinos, T. Roggendorf, R. Pfeifer and R. Wehner, "Insect strategies of visual homing in mobile robots", in *Biorobotics – Methods and Applications*, B. Webb and T. Consi, Eds, AAAI Press/MIT Press, 2001, pp. 37–66.

R. Nelson, "Visual homing using associative memory", in *Proceedings of the Image Understanding Workshop*, Palo CA: Morgan Kaufmann, 1989, pp. 245–262.

R. Nelson and A. Aloimonos, "Finding motion parameters from spherical motion fields (or the advantage of having eyes in the back of your head)", *Biol. Cybernet.*, 58, pp. 261–273, 1988.

W. Ribi, "The structural basis of information processing in the visual system of the bee", in *Neurobiology and Behavior of Honeybees*, R. Menzel and A. Mercer, Eds, Berlin: Springer, 1987, pp. 130–140.

A. Rizzi, D. Duina, S. Inelli and R. Cassinis, "A novel visual landmark matching for a biologically inspired homing", *Pattern Recognit. Lett.*, 22, pp. 1371–1378, 2001.

T. Röfer, "Controlling a wheelchair with image-based homing", in *Proceedings of AISB Workshop on Spatial Reasoning in Mobile Robots and Animals*, Manchester, 1997.

D. Ruderman and W. Bialek, "Statistics of natural images: scaling in the woods", *Phys. Rev. Lett.*, 73, pp. 814–817, 1994.

S. Se, D. Lowe and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks", *Int. J. Robot. Res.*, 21, pp. 735–758, 2001.

J. Stewart, *Calculus: Early Transcendentals*, 2nd edn, Pacific Grove, CA: Brooks/Cole, 1991.

N. Strausfeld, *Atlas of an Insect Brain*, Berlin: Springer, 1976.

W. Stürzl and H. Mallot, "Vision-based homing with a panoramic stereo sensor", in *Biologically Motivated Computer Vision, Lecture Notes in Computer Science 2525*, Berlin: Springer, 2002, pp. 620–628.

E. Todt and C. Torras, "Detecting salient cues through illumination-invariant color ratios", *Robot. Auton. Syst.*, 48, pp. 111–130, 2004.

E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Englewood Cliffs, NJ: Prentice Hall, 1998.

A. van der Schaaf and J. van Hateren, "Modelling the power spectra of natural images: statistics and information", *Vision Res.*, 36, pp. 2759–2770, 1996.

A. Vardy, "Biologically plausible methods for robot visual homing", PhD thesis, Carleton University (2005).

A. Vardy and F. Oppacher, “Low-level visual homing”, in *Advances in Artificial Life – Proceedings of the 7th European Conference on Artificial Life (ECAL), Vol. 2801 of Lecture Notes in Artificial Intelligence*, W. Banzhaf, T. Christaller, P. Dittrich, J.T. Kim and J. Ziegler, Eds, Berlin: Springer, 2003, pp. 875–884.

A. Vardy and F. Oppacher, “Anatomy and physiology of an artificial vision matrix”, in *Biologically Inspired Approaches to Advanced Information Technology: First International Workshop, BioADIT 2004, Vol. 3141 of Lecture Notes in Computer Science*, A. Ijspreet, M. Murata and N. Wakamiya, Eds, Berlin: Springer, 2004, pp. 290–305.

A. Vardy and F. Oppacher, “A scale invariant local image descriptor for visual homing”, in *Biomimetic Neural Learning for Intelligent Robots*, *vol. 3575 of Lecture Notes in Artificial Intelligence*. S. Wermter, G. Palm and E. Elshaw Eds. Berlin: Springer, 2005.

B. Webb, “Can robots make good models of biological behaviour?” *Behav. Brain Sci.*, 24, pp. 1033–1050, 2001.

K. Weber, S. Venkatesh and M. Srinivasan, “Insect-inspired robotic homing”, *Adaptive Behav.*, 7, pp. 65–97, 1999.

R. Wehner, B. Michel and P. Antonsen, “Visual navigation in insects: coupling of egocentric and geocentric information”, *J. Exp. Biol.*, 199, pp. 129–140, 1996.

J. Zar, *Biostatistical Analysis*, Englewood Cliffs, NJ: Prentice Hall, 1996.

J. Zeil, M. Hofmann and J. Chahl, “Catchment areas of panoramic snapshots in outdoor scenes”, *J. Opt. Soc. Am. A*, 20, pp. 450–469, 2003.

## Appendix A: Differential methods

### *A.1. FirstOrder*

Let $S$ and $C$ represent the snapshot and current respectively. Images, $S(x, y)$ and $C(x, y)$ denote the intensity of these images at coordinates $(x, y)$. The assumption that the intensity is unchanged under an image translation $(u, v)^T$ between the two images is expressed by

$$C(x+u, y+v) = S(x, y). \tag{A1}$$

Following the standard approach to differential techniques reviewed by Barron *et al.* (1994), we apply a first-order Taylor series expansion on the left-hand side of this equation to obtain

$$C(x, y) + [C_x(x, y), C_y(x, y)] \begin{pmatrix} u \\ v \end{pmatrix} = S(x, y), \tag{A2}$$

with $C_x$ and $C_y$ denoting partial derivatives with respect to $x$ and $y$. This Taylor series approximation is valid only for small translations.

With arguments $x$, $y$ omitted, we can write

$$(C_x, C_y) \begin{pmatrix} u \\ v \end{pmatrix} = S - C. \tag{A3}$$

Here we have two unknowns, $u$ and $v$, constrained by only one equation. Thus, there are infinitely many solutions. Therefore, we add the following constraint

$$\begin{pmatrix} u \\ v \end{pmatrix} = k \begin{pmatrix} C_x \\ C_y \end{pmatrix}. \tag{A4}$$

This constraint implies that the unknown correspondence vector $(u\ v)^T$ is aligned with the image gradient. It is a well-known problem that purely local differential techniques can estimate only that component of the correspondence vector which is parallel to the image gradient. This is known as the *aperture problem* (Trucco and Verri 1998: 192). We simply accept that the vector $(u\ v)^T$ will be only that component of the true correspondence vector that is parallel to the gradient.

We can solve for the constant $k$ by combining equations (A3) and (A4)

$$k = \frac{S - C}{\left\| \begin{pmatrix} C_x \\ C_y \end{pmatrix} \right\|^2}. \tag{A5}$$

Incorporating this into equation (A4) yields

$$\begin{pmatrix} u \\ v \end{pmatrix} = (S - C) \frac{\begin{pmatrix} C_x \\ C_y \end{pmatrix}}{\left\| \begin{pmatrix} C_x \\ C_y \end{pmatrix} \right\|^2}. \tag{A6}$$

It proved to be sufficient to ignore the length of this vector; therefore we use

$$\begin{pmatrix} u \\ v \end{pmatrix} = \operatorname{sgn}(S - C) \frac{\begin{pmatrix} C_x \\ C_y \end{pmatrix}}{\left\| \begin{pmatrix} C_x \\ C_y \end{pmatrix} \right\|^2} \tag{A7}$$

(for a zero gradient vector, the correspondence vector is set to zero). The function sgn(·) delivers the sign of its argument:

$$\operatorname{sgn}(x) = \begin{cases} -1; & x < 0 \\ 0; & x = 0 \\ 1; & x > 0. \end{cases} \tag{A8}$$

For the discrete approximation of the first-order derivatives, any first-order edge filter kernel can be applied to the image. Here we use the same kernels that were described in section 5: $(1\,0\,-1)$ and $(1\,0\,-1)^T$.

### *A.2. SecondOrder*

The second-order method rests on the assumption that the intensity gradient is preserved under translations (Barron *et al.* 1994), which can be expressed as

$$C_x(x + u, y + v) = S_x(x, y) \tag{A9}$$

$$C_y(x + u, y + v) = S_y(x, y). \tag{A10}$$

From a Taylor expansion on the left-hand side and approximation for small translations we obtain

$$C_x + (C_{xx}, C_{xy}) \begin{pmatrix} u \\ v \end{pmatrix} = S_x \tag{A11}$$

$$C_y + (C_{yx}, C_{yy}) \begin{pmatrix} u \\ v \end{pmatrix} = S_y, \tag{A12}$$

(with $C_{xx}$, $C_{xy}$, $C_{yx}$ and $C_{yy}$ denoting second-order partial derivatives) which can be summarized in a single equation (Barron *et al.* 1994):

$$\begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} D_x \\ D_y \end{pmatrix}. \tag{A13}$$

Here we use $D = S - C$. Note that $C_{xy} = C_{yx}$ holds for continuous derivatives. If the determinant of this matrix is zero then we have again encountered the aperture problem. This occurs for SecondOrder when the intensity is essentially one-dimensional. For the non-singular case, we obtain the solution.

$$u = \frac{D_x C_{yy} - D_y C_{xy}}{C_{xx} C_{yy} - C_{yx} C_{xy}} \tag{A14}$$

$$v = \frac{-D_x C_{yx} + D_y C_{xx}}{C_{xx} C_{yy} - C_{yx} C_{xy}}. \tag{A15}$$

We found that it was sufficient to consider only the sign of the components:

$$u = \mathrm{sgn}(D_x C_{yy} - D_y C_{xy})\mathrm{sgn}(C_{xx} C_{yy} - C_{yx} C_{xy}) \tag{A16}$$

$$v = \mathrm{sgn}(-D_x C_{yx} + D_y C_{xx})\mathrm{sgn}(C_{xx} C_{yy} - C_{yx} C_{xy}). \tag{A17}$$

With these equations, we obtain a zero vector in the singular case. Second-order derivatives are approximated by two subsequent convolutions of the image with the filter kernels described above.

## Appenidx B: Vector mapping methods

### *B.1. Exact solution*

We proceed from equation (14) to develop an expression for $\beta$. First we use the fact that $\sin(\pi - a) = \sin a$. This enables us to simplify the left-hand side of equation (14) slightly

$$\frac{\sin(\delta_x + \theta_x - \alpha)}{d_\mathbf{s}} = \frac{\sin(\theta_x - \alpha)}{d_\mathbf{c}} \tag{B1}$$

$$\frac{d_\mathbf{s}}{d_\mathbf{c}} = \frac{\sin(\delta_x + \theta_x - \alpha)}{\sin(\theta_x - \alpha)}. \tag{B2}$$

Next we use the addition rule for sines

$$\frac{d_\mathbf{s}}{d_\mathbf{c}} = \frac{\sin\delta_x \cos(\theta_x - \alpha) + \cos\delta_x \sin(\theta_x - \alpha)}{\sin(\theta_x - \alpha)} \tag{B3}$$

and factor out $\cos(\theta_x \alpha)$

$$\frac{d_\mathbf{s}}{d_\mathbf{c}} = \frac{\sin\delta_x + \cos\delta_x \tan(\theta_x - \alpha)}{\tan(\theta_x - \alpha)}. \tag{B4}$$

After a couple of further algebraic steps we obtain

$$\tan(\theta_x - \alpha) = \frac{\sin\delta_x}{\frac{d_\mathbf{s}}{d_\mathbf{c}} - \cos\delta_x} \tag{B5}$$

$$\theta_x - \alpha = \arctan\left(\frac{\sin\delta_x}{\dfrac{d_s}{d_c} - \cos\delta_x}\right). \tag{B6}$$

We use the fact that $\beta = \alpha + \pi$ to compute $\beta$

$$\beta = \theta_x - \arctan\left(\frac{\sin\delta_x}{\dfrac{d_\mathbf{s}}{d_\mathbf{c}} - \cos\delta_x}\right) + \pi. \tag{B7}$$

Finally, we include the result of equation (12) to obtain the final expression for $\beta$ in terms of $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$

$$\beta = \theta_x - \arctan\left(\frac{\sin\delta_x}{\frac{\tan(\theta_y+\delta_y)}{\tan\theta_y} - \cos\delta_x}\right) + \pi. \tag{B8}$$

### *B.2. Approximation*

As FirstOrder and SecondOrder only estimate the direction, not the distance, of correspondence it is not possible to use the exact vector mapping method. Therefore, a new approximate vector mapping technique, is developed in this section.

In the robot coordinate system $\mathbf{x}_\mathrm{R}$, $\mathbf{y}_\mathrm{R}$, the individual home vector $\mathbf{v}_\mathrm{h}$ can be expressed as the sum of the vector from **c** to **l** and the vector from **l** to **s**:

$$\mathbf{v}_h = d_\mathbf{c}\begin{pmatrix}\cos(\theta_x+\delta_x)\\ \sin(\theta_x+\delta_x)\end{pmatrix} - d_\mathbf{s}\begin{pmatrix}\cos\theta_x\\ \sin\theta_x\end{pmatrix}. \tag{B9}$$

In the approximated vector mapping technique, the individual home vector is expressed in coordinates $(\rho, \tau)$ with respect to a coordinate system $(\mathbf{v}_r, \mathbf{v}_t)$ which is attached to the horizontal angle where the landmark **l** appears in the snapshot (see figure 20):

$$\mathbf{v}_h = \rho\mathbf{v}_r + \tau\mathbf{v}_\mathrm{t}. \tag{B10}$$

With the radial and tangential coordinate vector

$$\mathbf{v}_r = \begin{pmatrix}\cos\theta_x\\ \sin\theta_x\end{pmatrix}, \quad \mathbf{v}_t = \begin{pmatrix}\cos\left(\theta_x+\dfrac{\pi}{2}\right)\\ \sin\left(\theta_x+\dfrac{\pi}{2}\right)\end{pmatrix} = \begin{pmatrix}-\sin\theta_x\\ \cos\theta_x\end{pmatrix} \tag{B11}$$

we obtain the following equation for the first vector component:

$$\rho\cos\theta_x - \tau\sin\theta_x = d_\mathbf{c}\cos(\theta_x+\delta_x) - d_\mathbf{s}\cos\theta_x. \tag{B12}$$

After applying an addition theorem to the right-hand side and rearranging terms we arrive at

$$\rho\cos\theta_x - \tau\sin\theta_x = (d_\mathbf{c}\cos\delta_x - d_\mathbf{s})\cos\theta_x - d_\mathbf{c}\sin\delta_x\sin\theta_x \tag{B13}$$

which, by comparing the coefficients, gives

$$\rho = d_\mathbf{c}\cos\delta_x - d_\mathbf{s} \tag{B14}$$

$$\tau = d_\mathbf{c}\sin\delta_x. \tag{B15}$$

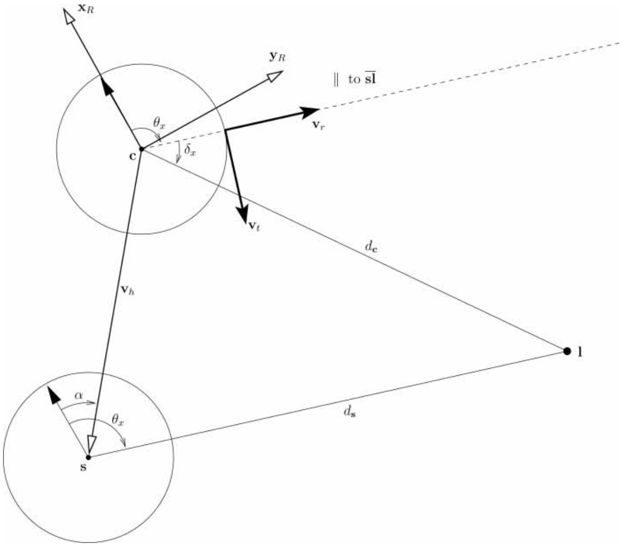


Figure 20. Illustration of the robot coordinate system and the vectors involved in the approximate vector mapping technique.

We are interested only in preserving the ratio of $\rho$ and $\tau$. By factoring out $d_\mathbf{c}$ and approximating $\cos\delta_x \approx 1$ and $\sin\delta_x \approx \delta_x$ for small $\delta_x$, we obtain

$$\rho' = 1 - \frac{d_\mathbf{s}}{d_\mathbf{c}} \tag{B16}$$

$$\tau' = \delta_x. \tag{B17}$$

By using equation (12), we can continue for $\rho'$ with

$$\rho' = 1 - \frac{\tan(\theta_y + \delta_y)}{\tan\theta_y}. \tag{B18}$$

For small vertical translations $\delta_y$, we can approximate $\tan(\theta_y + \delta_y)$ by a Taylor expansion up to first-order terms:

$$\rho' = 1 - \frac{\tan\theta_y + \delta_y \tan'\theta_y}{\tan\theta_y} = \frac{-\delta_y}{\tan\theta_y \cos^2\theta_y} = \frac{-\delta_y}{\sin\theta_y \cos\theta_y}. \tag{B19}$$

For vertical landmark angles $\theta_y$ which stay close to the horizon, we obtain the following approximation:

$$\rho' = -\frac{\delta_y}{\theta_y}. \tag{B20}$$

Instead of using the value of $\theta_y$, we observed that considering only its sign leads to better homing performance, which gives

$$\rho' = -\mathrm{sgn}(\theta_y)\delta_y \tag{B21}$$

$$\tau' = \delta_x. \tag{B22}$$

Finally, the individual home vector is normalized to unit length:

$$\rho'' = \frac{-\operatorname{sgn}(\theta_y)\delta_y}{\sqrt{\delta_x^2 + \delta_y^2}} \tag{B23}$$

$$\tau'' = \frac{\delta_x}{\sqrt{\delta_x^2 + \delta_y^2}} \tag{B24}$$

A slight gain in performance is obtained for FirstOrder if the home vector is subsequently multiplied by the length of the gradient $\|(C_x C_y)^T\|$. This is because the flow field can best be approximated by differential techniques where the image gradient is strongest (see Trucco and Verri 1998: chapter 8).