Auton Robot (2007) 23: 183–196
DOI 10.1007/s10514-007-9040-0

Metric embedding of view-graphs

A vision and odometry-based approach to cognitive mapping

Wolfgang Hübner · Hanspeter A. Mallot

Received: 24 April 2006 / Accepted: 4 May 2007 / Published online: 22 June 2007

Abstract Most recent robotic systems, capable of exploring unknown environments, use topological structures (graphs) as a spatial representation. Localization can be done by deriving an estimate of the global pose from landmark information. In this case navigation is tightly coupled to metric knowledge, and hence the derived control method is mainly pose-based. Alternative to continuous metric localization, it is also possible to base localization methods on weaker constraints, e.g. the similarity between images capturing the appearance of places or landmarks. In this case navigation can be controlled by a homing algorithm. Similarity based localization can be scaled to continuous metric localization by adding additional constraints, such as alignment of depth estimates.

We present a method to scale a similarity based navigation system (the view-graph-model) to continuous metric localization. Instead of changing the landmark model, we embed the graph into the three dimensional pose space. Therefore, recalibration of the path integrator is only possible at discrete locations in the environment. The navigation behavior of the robot is controlled by a homing algorithm which combines three local navigation capabilities, obstacle avoidance, path integration, and scene based homing. This homing scheme allows automated adaptation to the environment. It is further used to compensate for path integration errors, and therefore allows to derive globally consistent pose estimates based on "weak" metric knowledge, i.e. knowledge solely derived from odometry. The system performance is tested with a robotic setup and with a simulated agent which explores a large, open, and cluttered environment.

Keywords Autonomous map building · Simultaneous localization and mapping · Cognitive mapping

This work is part of the GNOSYS (FP6-003835-GNOSYS) project, supported by the European Commission.

W. Hübner (✉) · H.A. Mallot
Department of Zoology, Cognitive Neuroscience, Auf der Morgenstelle 28, 72076 Tübingen, Germany
e-mail: wolfgang.huebner@uni-tuebingen.de

H.A. Mallot
e-mail: hanspeter.mallot@uni-tuebingen.de

1 Introduction

Metric information, in terms of the robot position and heading (pose), plays a central role in most recent robotic systems, capable of exploring and mapping unknown environments. Acquisition of spatial knowledge is usually done by consistently integrating landmark positions into a global reference frame. The use of metric information in these systems differs from the use of metric knowledge in biologically inspired systems in two respects. Firstly, pose information in biological systems only has a supporting role. Evidence for this idea could be seen from experiments with rat navigation, where it has been shown that place fields can be established without input from a path integration system (odometry), i.e. navigation is not purely pose controlled (Redish 1999). A supporting role of metric has further been suggested in the spatial semantic hierarchy, where metric knowledge is the highest level in a hierarchy of spatial knowledge (survey knowledge) and not its basis (Kuipers 2000).

A second difference is that precise metric knowledge evolves slowly over time, while technical systems are build to achieve an acceptable level of confidence quite fast. Evidence for this could be seen from a long term experiment

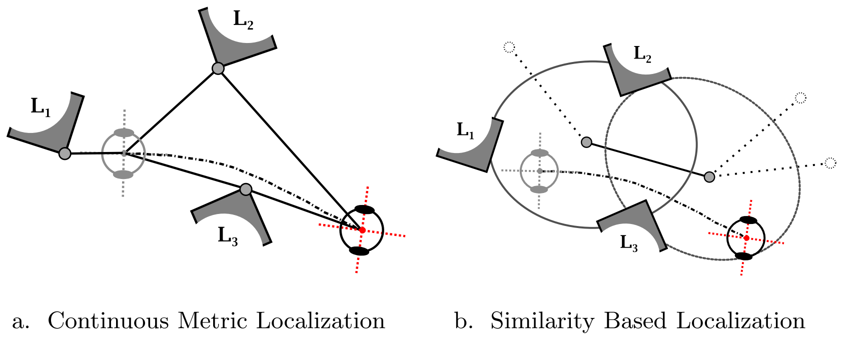
L1
L2
L3
a. Continuous Metric Localization
b. Similarity Based Localization

Fig. 1 Illustration of two different localization methods. **a** The robot is able to continuously determine an estimate about its pose while moving along a trajectory (*dashed black curve*). Uncertainty in the pose estimate is determined by the number of integrated landmarks (*gray dots*) and the sensor used to generate the landmark model. **b** Instead of using pose estimates, localization can also be based on the similarity between views. In our system views represent the appearance of places. Therefore, the robot is only able to determine a rough indication of how close (in terms of image similarity) it is to a known location

described in Golledge and Stimson (1997). This study addressed the metric knowledge of people, who had newly arrived in an unknown city, and therefore were totally unfamiliar with the new environment. In order to monitor the gain of spatial knowledge the spatial layout of the cognitive map has been reconstructed using a multidimensional scaling (MDS) algorithm (Borg and Groenen 1997). The learning progress, which has been observed over nine months, showed a steadily increasing precision of metric knowledge. It has further been shown that subjects tend to stick to preferred routes, and that these routes are represented more accurately than less experienced routes. Results from this long term study also show similarities with the convergence of a relaxation based mapping method (Duckett et al. 2002). The fact that human subjects show robust navigation capabilities also during learning further illustrates that navigation does not necessarily depend on consistent metric knowledge.

In order to model the complex task of cognitive mapping with autonomous robots, the first problem is to define a localization method. The localization method determines the model used to represent the environment, as well as the algorithm used to generate the robots navigation behavior (Kuipers 2000; Mallot 1999).

Localization. Figure 1(a) illustrates the method of *continuous metric localization*. In this approach the location is determined by continuously integrating landmark observations and egomotion estimates into an estimate about the instantaneous pose and its uncertainty (e.g. Thrun et al. 2004). Hence, the accuracy of localization is constrained by the number of integrated landmarks and by the accuracy of the sensor used to derive landmark positions. Alternatively, localization can also be based on "weaker" constraints, which is illustrated in Fig. 1(b). In its simplest form, localization is only based on a measure of closeness to a known location, e.g. the similarity between a stored image and the instantaneously perceived image (e.g. Franz et al. 1998b; Gourichon et al. 2002; Stürzl and Mallot 2002). This localization method is closely related to the notion of a place code, used in the context of place cell models (Redish 1999). It is further related to the snapshot model, which has been used to explain navigation capabilities of honey bees (Cartwright and Collett 1983) and desert ants (Wehner et al. 1996; Lambrinos et al. 2000).

Spatial Representation. In contrast to early robotic systems, where the environment has been represented on a regular grid, most recent spatial representations make use of a sample pattern that is adapted to the landmark density. One possibility is to represent a map as a set of landmarks and their positions. Uncertainty about landmark positions can be represented explicitly in the co-variance matrix or alternatively implicitly in the information matrix, which has further been used to define efficient mapping algorithms (Thrun et al. 2001; Menegatti et al. 2004; Liu and Thrun 2003; Thrun et al. 2004). A second possibility is to use graph models, where the node set of the graph represents known places or landmarks according to the applied landmark model. This method can be applied totally without metric (Franz et al. 1998a). Graph models can also be scaled in order to support continuous metric localization. Usually this is done by adding additional constraints, i.e. the landmark model must contain depth estimates. The depth scans are pairwise aligned in order to derive local metric relations. The final step, i.e. integration of local metric relations into a globally consistent map is done by an optimization method (Lu and Milios 1997; Gutmann and Konolige 1999).

Action Generation. In case of continuous metric localization the robot motion can always be controlled in terms of a target pose. The same can be done, if the geometric structure of the environment is accurately sampled, i.e. if the landmark model includes depth estimates and the depth scans are properly aligned (Lu and Milios 1997; Gutmann and Konolige 1999; Duckett et al. 2002) or in case pose estimates are derived directly from vision (Sim and Dudek 1998; Se et al. 2005). Alternatively, motion can be controlled by a homing algorithm,[1] derived from the landmark model. In other words, as an alternative to a fully pose-based control scheme a selforganized controller is used, which works independent of the metric components of the map. Although visual homing depends on several parameters which have to be adjusted to the environment, it can be shown that the overall homing performance is not very sensitive to these parameters. Therefore, homing strategies have become a popular tool for robot localization (Argyros et al. 2001; Möller et al. 2001; Hafner and Möller 2001; Cassinis et al. 2002; Stürzl and Mallot 2002; Vardy and Möller 2005; Franz et al. 1998b).

In this article we introduce a method to scale the view graph model to continuous metric localization. Instead of constraining the landmark model to a metrically complete representation, we only label each node with a pose estimate. Therefore, each node in the graph supports a global pose estimate which is used as a local reference frame for path integration. Motion outside the graph mesh is done only with path integration. We will show that this approach allows to learn globally consistent maps, based only on weak metric knowledge derived solely from odometry. The major difference to other mapping systems is that the system allows to derive metric knowledge without using a sensor that is able to give distance information.

As pointed out in the introduction most components of the system (especially the single navigation behaviors) are motivated from biological examples. Therefore, the full system is considered as a model of cognitive mapping. Nevertheless, most components have an equivalent counterpart in engineering, e.g. the visual homing method, which is referred to as visual serving in the technical literature (e.g. Hutchinson et al. 1996). We suggest, however, that the presented mapping algorithm will be useful as a technical solution in four respects. First, the approach offers a great flexibility in the choice of the landmark model (and sensors), as long as the landmark model allows the definition of a homing algorithm. Second, the approach is robust and designed for applications in cluttered or outdoor environments,[2] which are hard to map with Laser-based approaches. Third, the approach is cheap, both in terms of hardware requirements and computation. Finally, it suggests a modularization of mapping components which we think is practically efficient.

The article is structured as follows. Section 2 gives a brief overview of the local navigation strategies, upon which the more complex exploration capabilities are build. Section 3 describes the method used to derive globally consistent pose estimates. Section 4 describes the exploration algorithm and illustrates the resulting behavior. Section 5 shows results, illustrating the system performance in a robotic setup and with a simulation of a large, open, and cluttered environment.

[1]The application of a homing algorithm is equal to the hill climbing step, used in the spatial semantic hierarchy (Kuipers 2000).

[2]Extensions to outdoor environments are subject to ongoing work.

2 Local navigation strategies

Local navigation is defined as navigation within the sensory horizon, i.e. within a visible vicinity of a goal location. Global navigation, which will be described in the next section, is the extension of local navigation by representing multiple locations together with their spatial relations in a map. The basis for our agent is built of three local navigation strategies: *obstacle avoidance*, *scene based homing*, and *path integration*, which will be described in the following.

Obstacle Avoidance. Our system is restricted to open, static environments. Therefore, we use a simple reactive and hardwired obstacle avoidance mechanism, based on a lookup table, directly linking IR-sensor readings to predefined motor responses. The resulting behavior ensures, that the agent faces open space, without moving too far from the collision point. This scheme is comparable to a Braitenberg-vehicle (Braitenberg 1984). In order to avoid deadlock situations, the duration of an avoidance sequence is randomized.

Scene Based Homing. This method describes the ability of returning to a known place by comparing two views, the instantaneous view ($\boldsymbol{I}_t$) and the view recorded at the home or "recording" location ($\boldsymbol{I}^h$). The most prominent model in this context is the *snapshot model* (Cartwright and Collett 1983), which has also been used as the basis for several robotic implementations (e.g. Röfer 1997; Floreano and Mondada 1996; Franz et al. 1998b; Weber et al. 1999; Möller 2000; Argyros et al. 2001; Hafner and Möller 2001; Möller et al. 2001; Rizzi et al. 2001; Cassinis et al. 2002; Gourichon et al. 2002; Stürzl and Mallot 2002; Vardy and Oppacher 2003; Vardy and Oppacher 2004; Vardy and Möller 2005).

In order to capture the perceived landmark configuration, a 360 degree field of view is advantageous, since all image features are kept in the visual field, irrespective of the motion. We follow the method described in Franz et al. (1998b), where a snapshot consists of 72 gray value pixels ($\boldsymbol{I}_t \in [0, 255]^{72}$) extracted from the horizon line of the panoramic view.

In order to derive a motion decision from snapshots, it is necessary to define a measure of similarity between the instantaneously perceived view ($\boldsymbol{I}_t$) and the stored home view ($\boldsymbol{I}^h$). This could be done applying global search methods, like the maximum cross-correlation or the minimum sum of squared differences (e.g. Jähne 1999). In more recent publications also local search methods, such as attractor like networks (Vardy and Oppacher 2003), optic flow methods (Vardy and Möller 2005) or scale invariant image descriptors (Vardy and Oppacher 2004) have been proposed.

The result of matching two snapshots is a motion decision, minimizing the expected dissimilarity between both views.

$$\boldsymbol{p}_t^h(\boldsymbol{I}^h, \boldsymbol{I}_t) = (\rho_t^h, d_t^h)^\top = \arg\min_{\rho,d} (\boldsymbol{I}^h - \boldsymbol{I}^c(\boldsymbol{I}_t, \rho, d))^2. \quad (1)$$

Motion decisions are coded as egocentric polar vectors $\boldsymbol{p}_t^h = (\rho_t^h, d_t^h)^\top$. The polar vector describes a rotation (ρ_t^h) followed by a translation (d_t^h). $\boldsymbol{I}^c(\boldsymbol{I}_t, \rho, d)$ is the expected snapshot at the relative location (ρ, d), calculated with the *image warping algorithm* (Franz et al. 1998b). Iteration of prediction and motion results in a full homing trial, which is given as a sequence of rotations and translations $S^h = \{\boldsymbol{p}_0^h, \boldsymbol{p}_1^h, \dots, \boldsymbol{p}_n^h\}$. The iteration successfully ends if $\boldsymbol{p}_n^h \approx (\rho_n^h, 0)^\top$, i.e. if the agent could no longer proceed closer to the home location by image matching. If this is not the case after a fixed number of iterations, it is assumed that the homing trial has failed.

The following three quantities are used to characterize the performance of the homing algorithm.

1. The *Catchment-Area* is defined as the area around a recording location, where the visual homing method does succeed.

$$C(\boldsymbol{I}^h) = \{\boldsymbol{P}^w | \exists S : \boldsymbol{P}_h^w - ((\boldsymbol{P}^w \oplus \boldsymbol{p}_0^h) \oplus \cdots \oplus \boldsymbol{p}_n^h) \le \epsilon\} \quad (2)$$

$\boldsymbol{P}^w$ defines the robots pose in a world coordinate system, $\boldsymbol{P}_h^w$ is the pose of the home snapshot. The operator $\oplus$ defines the transition of the pose vector according to a motion decision

$$\boldsymbol{P}_{t+1} = \boldsymbol{P}_t \oplus \boldsymbol{p}_t = \begin{pmatrix} x_t \\ y_t \\ \phi_t \end{pmatrix} + \begin{pmatrix} d_t \cos(\phi_t + \rho_t) \\ d_t \sin(\phi_t + \rho_t) \\ \phi_t + \rho_t \end{pmatrix}. \quad (3)$$

2. *Homing-Range.* Since size and shape of the catchment-area are unknown to the agent, it is necessary to use a place recognition system. The place recognition system is responsible for determining possible goal locations, which amounts to the decision whether the agent is located inside or outside of a catchment-area, i.e. to predict whether a homing trial will succeed or not. For this purpose we use the following rotation invariant measure of image similarity:[3]

$$d_{\text{corr}}(\boldsymbol{I}_1, \boldsymbol{I}_2) = \max_{\phi_s \in (0,2\pi)} (\boldsymbol{I}_1 \circ \boldsymbol{I}_2(\phi_s)). \quad (4)$$

[3] $\boldsymbol{I}(\boldsymbol{P}^w)$ defines the snapshot perceived at pose $\boldsymbol{P}^w$. $\boldsymbol{I}(\phi)$ defines a rotated snapshot, which corresponds to a change in the robot's orientation about ϕ. $\boldsymbol{I}_1 \circ \boldsymbol{I}_2$ defines the inner product between the two image vectors.

A homing trial is assumed to be promising if two views are sufficiently similar, i.e. if $d_{\text{corr}}(\boldsymbol{I}_t, \boldsymbol{I}^h) > \text{const}$. The homing range of a snapshot is defined as the set of all possible locations, where a successful homing trial is predicted:

$$H(\boldsymbol{I}^h) = \{\boldsymbol{P}^w | d_{\text{corr}}(\boldsymbol{I}^h, \boldsymbol{I}(\boldsymbol{P}^w)) > \text{const}\}. \quad (5)$$

In comparison to continuous metric localization, this is a less precise localization method, since (5) does not allow to specify the agent's exact position within the catchment area, be it in terms of a global or a local pose. Further problems arise, since H is not guaranteed to be a subset of C, resulting in homing trials which may fail. Furthermore the problem of *spatial aliasing* arises, since H is also not guaranteed to form one connected region.

3. *Region of uncertainty.* The region where the agent cannot not proceed closer to the home location by matching snapshots,

$$U(\boldsymbol{I}^h) = \{\boldsymbol{P}^w | \boldsymbol{p}_t^h(\boldsymbol{I}^h, \boldsymbol{I}(\boldsymbol{P}^w)) \approx (\rho, 0)^\top\} \quad (6)$$

defines the spatial resolution of the algorithm.

The relation between the three quantities can be described using the local image variation (*liv*), i.e. the variation in the perceived view $\boldsymbol{I}$ according to a change in position (x, y).

$$\text{liv}^2(\boldsymbol{x}) = \det\left[\sum_\phi \boldsymbol{g}\boldsymbol{g}^\top\right], \qquad \boldsymbol{g} = \left(\frac{\partial I(\boldsymbol{P}^w)}{\partial x}, \frac{\partial I(\boldsymbol{P}^w)}{\partial y}\right)^\top. \quad (7)$$

It has been shown (Stürzl 2004) that the size of the catchment-area is inversely proportional to the image variation and directly proportional to the region of uncertainty. An increase in the image variation, i.e. a large liv-value results in a decreased catchment-area size and in an increased spatial resolution. In summary, the performance of a homing algorithm is a function of the recording location and is therefore independent of the agent's history.

Path Integration, derived from odometry, is a common method to achieve metric knowledge. Usually, the state of the path integrator is defined as a probability distribution with the instantaneous pose vector $\boldsymbol{P}_t = (x_t, y_t, \phi_t)^\top$ as the mean and the co-variance matrix $C_t \in \mathbb{R}^{3\times 3}$ representing the uncertainty about the unknown true state ($\boldsymbol{P}_t^w$).

The path integration task is to keep track of the instantaneous pose, estimated relative to an arbitrary starting point $\boldsymbol{P}_0 = (0, 0, 0)^\top$. Updates of $\boldsymbol{P}_t$ are calculated from egomotion estimates $\boldsymbol{d}_t = (d_t, \rho_t)$. In order to account for error accumulation in (3) we apply the probabilistic odometry model described in Thrun et al. (2005). Contrary to scene based homing, uncertainty in path integration is a function of the trajectory and is therefore independent of the location. Figure 2 illustrates the situation for a homing by path

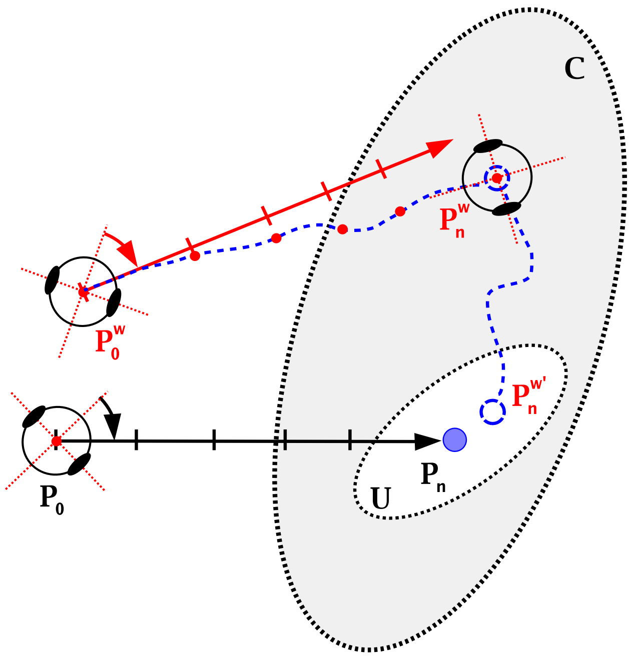

Fig. 2 Combined homing scheme used to correct position errors from path integration, using scene based homing

integration task. The intended course (black arrow) is calculated between an uncertain start location ($\boldsymbol{P}_0$) and the uncertain goal location ($\boldsymbol{P}_n$). Due to the erroneous start condition, the motion decision leads the robot along the red arrow. According to the motion model, the robot is not able to follow precisely the chosen direction causing deviations from the intended course (dashed blue line). Thus the robot ends up at $\boldsymbol{P}^w_n$.

Following the example of ant navigation (Wehner et al. 1996), we use a combination of scene based homing and homing by path integration in order to define a more robust homing scheme. If homing by path integration ends up at $\boldsymbol{P}^w_n$ and the perceived view at $\boldsymbol{P}^w_n$ is sufficiently distinctive from the view recorded at the goal location, it is possible to correct the position error using scene based homing (illustrated as the curved dashed trajectory in Fig. 2). Here the phrase "sufficiently distinctive" means that the agent must be located inside C and outside U. A small position error remains, since the scene based homing algorithm terminates as soon as the robot enters U.

As will be described in more detail in the next section, this method is different from recent robotic systems. Although the visual correction guides the robot closer to the home location, the estimate of the total pose change between the start location and the goal location becomes more and more uncertain, because the trajectory becomes longer. Therefore the system could not benefit from several independent sensors, in order to reduce path integration errors. Further, this shows how all local metric relations in our system are based on measurements derived from odometry. According to the distinction between weak and strong links in (Lu and Milios 1997) our system only relies on weak links.

The combined homing scheme shows some similarities to Bayes-filters (like Kalman-filter, or particle filter) applied to the problem of state estimation. According to Thrun et al. (2005), motion induces a loss of information (certainty), while perception (update-cycle in terms of the Kalman-filter) induces a gain of information. This is similar to our system, since the combined homing scheme first generates a prediction of a target location which is updated in a second step by the visual homing mechanism. Compared to the application of Bayes-filters, perception of a snapshot in our system does not increase certainty about the location during the visual homing phase. An update of the pose estimate is only possible after a successful homing trial.

In order to integrate the three navigation capabilities (obstacle avoidance, visual homing, and homing by path integration) we use a subsumption schema (Brooks 1986). First the robot tests for obstacles in its way. If obstacles are detected the obstacle avoidance tries to move the robot back into open space. If the robot is repeatedly forced to avoid obstacles, the homing trial is aborted. If the robot moves in open space, it first follows its metric vector, and then switches to visual homing.

3 Large scale navigation

In previous work a robotic implementation of the *view graph model* (Franz et al. 1998a) has been introduced. In the view graph $G = (V, E)$, each node is labeled with a view of the environment ($V_i = \{\boldsymbol{I}_i\}$). The robot was able to move between two nodes using the visual homing algorithm described in the previous section. Edges are used to represent the agents experience during exploration. In order to ensure a secure navigation, an edge has been labeled as *traversable* if both nodes are located inside the catchment area of each other. Otherwise an edge is labeled as *non-traversable*. If no edge exists, no information about a possible route has been experienced.

In the following we extend the view graph model from topological information to survey knowledge by embedding the graph into the three dimensional pose space. Therefore, node labels are extended to $V_i = \{\boldsymbol{I}_i, \boldsymbol{P}_i\}$. $\boldsymbol{P}_i$ is an estimate of the node pose in the reference frame $\boldsymbol{P}_0 = (0, 0, 0)^\top$ and $\boldsymbol{P}_1 = (x_1, 0, 0)^\top$. Further, labels of traversable edges are augmented by a vector describing the pose change between two nodes, i.e. $E_{ij} = \{\Delta\boldsymbol{P}_{ij}\}$. Therefore, the graph only contains information about the location of landmarks if the landmark model includes depth estimates.

Pose Relation Network. Since path integration errors (see Fig. 3) lead to inconsistent global pose estimates, it is nec-

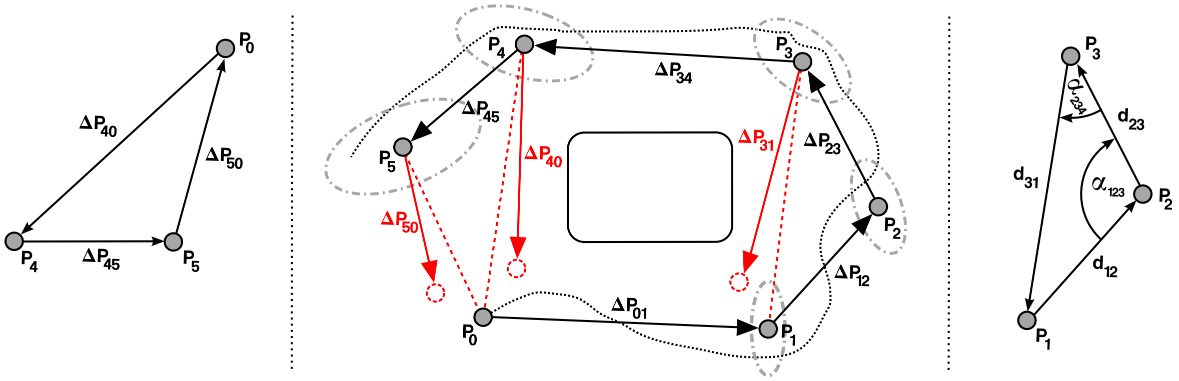

Fig. 3 Two methods used to remove inconsistencies in the pose relation network. Both methods minimize the mismatch between the local pose relations ($\Delta \boldsymbol{P}_{ij}$) and the global pose estimates ($\boldsymbol{P}_k$). The left side illustrates application of (9). The global node configuration is transformed into a local reference frame spawned by $\boldsymbol{P}_4$. The right side illustrates application of (10). Here, the mismatch is calculated directly by comparing angles and distances

essary to define the embedding ($f(V) \to \mathbb{R}^3$) as an optimization problem, which is usually defined by the following objective function:

$$Q(X, D) = \sum_{(i,j)\in E} \sum_m f(\boldsymbol{P}_j, \boldsymbol{P}_i, \Delta \boldsymbol{P}^m_{ij})^\top \times C^{-1}_{ij} f(\boldsymbol{P}_j, \boldsymbol{P}_i, \Delta \boldsymbol{P}^m_{ij}). \tag{8}$$

Function $f(.)$ measures the dissimilarity between the global node configuration $X = \{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_{|V|}\}$ and the local pose relations $D = \{\boldsymbol{\Delta P}^m_{ij}\}$. The co-variance matrix (C_{ij}) is taken from the motion model. Inconsistencies are removed by minimizing (8), i.e. $X_0 = \arg\min_X Q(X, D)$.

Matching pose vectors. Each local pose relation defines the pose change between two nodes. A dissimilarity measure for (8) can be defined by transforming the global pose estimates into a local reference frame, or vice versa (see left part of Fig. 3). The dissimilarity measure is then (Lu and Milios 1994, 1997; Duckett et al. 2002; Frese et al. 2004; Golfarelli et al. 1998) defined as:

$$f(\boldsymbol{P}_j, \boldsymbol{P}_i, \boldsymbol{\Delta P}_{ij}) = R(\phi_i)(\boldsymbol{P}_j - \boldsymbol{P}_i) - \boldsymbol{\Delta P}_{ij}. \tag{9}$$

If the robot is equipped with a compass (i.e. $R(\phi) \approx$ const) equation (9) becomes a system of linear equations, which makes things more easy.

Matching point configurations. Alternatively the dissimilarity function can be defined, using the mismatch of metric relations, such as distances and angles. Since distances and angles are invariant under rigid body transformation, there is no need to transform pose vectors from one reference frame to another. Such methods are referred to as *multidimensional-scaling* (MDS) algorithms (Borg and Groenen 1997; Mardia et al. 1982). Therefore, we use the dot-product and the cross-product of the position parts of two pose relations to account for position estimates and heading estimates. Further, application of the two products has the benefit that rotation estimates and distance estimates are automatically integrated. The original objective function (8) becomes:

$$Q_p(X, D) = \sum_{\{i,j,k\}} \sum_m^{|D_{ijk}|} \frac{1}{\sigma^2_{s^m_{ijk}}} [s(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_k - \boldsymbol{x}_j) - s(\Delta \boldsymbol{x}^m_{ij}, \Delta \boldsymbol{x}^m_{jk})]^2 + \frac{1}{\sigma^2_{d^m_{ijk}}} [d(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_k - \boldsymbol{x}_j) - d(\Delta \boldsymbol{x}^m_{ij}, \Delta \boldsymbol{x}^m_{jk})]^2 \tag{10}$$

$\{i, j, k\}$ is the set of triplets, $D_{i,j,k}$ is the set of all measurements taken along this triplet. The triplet measures the change of the robot's pose while the robot moves from node i to node j and further to node k, i.e. the pose of nodes j and k are measured in the reference frame of node i (see Fig. 3).

The dissimilarity measure is defined as:

$$s(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_k - \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j) \circ (\boldsymbol{x}_k - \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \|\boldsymbol{x}_k - \boldsymbol{x}_j\| \cos \alpha_{ijk}. \tag{11}$$

Using uncertainties from the motion model, the variance of the dot product is approximated by:

$$\sigma^2_{s_{ijk}} \approx (2x_j - x_k)^2 \sigma^2_{x_j} + x^2_j \sigma^2_{x_k} + (2y_j - y_k)^2 \sigma^2_{y_j} + y^2_j \sigma^2_{y_k} + 2(2x_j - x_k)(2y_j - y_k)\sigma_{x_j y_i} + 2x_j y_j \sigma_{x_k y_k}. \tag{12}$$

The cross product is used as a second similarity measure, in order to make the inner angle unique

$$d(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_k - \boldsymbol{x}_j) = \det(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_k - \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \|\boldsymbol{x}_k - \boldsymbol{x}_j\| \sin \alpha_{ijk}. \tag{13}$$

Again, the variance of the cross product is approximated by:

$$\sigma^2_{d_{ijk}} \approx y_k^2\sigma^2_{x_j} + x_k^2\sigma^2_{y_j} + y_j^2\sigma^2_{x_k} + x_j^2\sigma^2_{y_k} - 2x_k y_k \sigma_{x_j y_j} - 2x_j y_j \sigma_{x_k y_k}. \tag{14}$$

In order to avoid ambiguous solutions, it is necessary to include measurements of the form $\{i, j, i\}$ along an edge (i, j). Thus each edge is also modeled as two segments with a turning angle of 180°.

Heading Estimation. Since (10) does no longer depend on the global heading of the robot, it becomes possible to separate the calculation of globally consistent positions from the calculation of globally consistent orientations.

After a successful homing trial, the best match between the home snapshot ($\boldsymbol{I}^h$) and the instantaneous snapshot ($\boldsymbol{I}_t$) yields the heading difference between the recording heading (ϕ_r) and the instantaneous heading (ϕ_t):

$$\upsilon_t = (\phi_t - \phi_r) \bmod 2\pi = \arg \max_{\phi_s \in (0,2\pi)} (\boldsymbol{I}_t \circ \boldsymbol{I}^h(\phi_s)). \tag{15}$$

According to equation (15) it is possible to derive the instantaneous heading, if the recording heading is known. Further, the total heading change along an edge (E_{ij}) is given by:

$$\Delta\omega_{ji} = \sum_{k=1}^{n} \rho_k + \upsilon_i = \Delta\phi_{ji} + \upsilon_i \quad \text{with } (j, i) \in E. \tag{16}$$

Inconsistent heading estimates in the pose relation network are again solved by minimizing an objective function. Therefore, we search for the set of recording directions $\boldsymbol{\Phi}_r = \{\phi_{r1}, \ldots, \phi_{rn}\}$ which fits best into the set of local orientation changes $\boldsymbol{\Omega}$.

$$Q_r(\boldsymbol{\Phi}, \boldsymbol{\Omega}) = \sum_{(i,j)\in E} \sum_{m=0}^{|\boldsymbol{\Omega}_{ij}|} \frac{1}{\sigma^2_{\Delta\omega^m_{ij}}} \times [(\phi_{rj} - \phi_{ri} - \Delta\omega^m_{ij}) \bmod 2\pi]^2. \tag{17}$$

The error function is periodic which causes problems applying a gradient descent method. Therefore, it is more convenient to use unit vectors:

$$Q_r(\boldsymbol{\Phi}, \boldsymbol{\Omega}) = \sum_{(i,j)\in E} \sum_{m=0}^{|\boldsymbol{\Omega}_{ij}|} \frac{1}{\sigma^2_{\Delta\omega^m_{ij}}} (\|\boldsymbol{x}_{ij} - \boldsymbol{\Delta x}^m_{ij}\|)^2, \tag{18}$$

with

$$\boldsymbol{x}_{ij} = (\sin(\phi_{ri} - \phi_{rj}), \cos(\phi_{ri} - \phi_{rj}))^\top$$

and

$$\boldsymbol{\Delta x}^m_{ij} = (\sin \Delta\omega^m_{ij}, \cos \Delta\omega^m_{ij})^\top.$$

Pose Relations along an edge are directly derived from odometry. As can be seen from Fig. 2, the most certain measurements are achieved if the robot moves on a straight path between two nodes. During exploration it is possible that multiple measurements are recorded along the same edge, indicated by the index m in (10) and (18). It has been shown (Whaite and Ferrie 1997; MacKay 1992) that measurements recorded at locations where model predictions are most uncertain are most informative and therefore speed up convergence. Applied to the homing scheme we can assume that homing trials without a visual correction step are less informative, since the resulting measurement only varies according to odometry errors.

Iterative Map Updates. Since the complexity of minimizing objective functions (10) and (18) increases with the map size, it is necessary to use local map updates. In Duckett et al. (2002) and Frese et al. (2004) a relaxation method has been proposed as a solution to this problem. Here we use an adaption of such a relaxation scheme applied to our nonlinear model. Following this idea we rewrite the original objective function (10) in the following way:

$$Q(X_f, X_v, D') = \sum_{(i,j,k,m)\in D'} E(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k, \Delta\boldsymbol{x}^m_{ij}, \Delta\boldsymbol{x}^m_{jk})^2. \tag{19}$$

$E(.)$ is the same error term as in (10). $X_v \subseteq X \backslash X_f$ is a set of vertices for which new position estimates are calculated according to: $\Delta X_v = \arg\min_{X_v} Q(X_f, X_v, D')$. $X_f \subseteq X \backslash X_v$ is a set of fixed vertices, which build a reference frame for locally integrating the vertices X_v. Depending on the available local measurements, the choice of X_f could be determined by X_v, so that X_f consists of all vertices which can predict locations of vertices in X_v. D' is a subset of D, selected according to the vertices $X_v \cup X_f$. As applied here, X_v consists, in one iteration step, only of the node for which a new local pose relation has been recorded. X_f includes all first and second neighbors of X_v.

After position estimates for X_v have been updated, the above procedure is repeated for all neighbors of X_v which have been significantly moved. This iteration cycle is aborted if the total movement of vertices is below a certain threshold, i.e. if the gradient of objective function (10) has entries close to zero for all elements of X_v.

4 Exploration

In this section we describe the algorithm used to control the exploration behavior of the robot. As shown in Fig. 4 the exploration algorithm consists of three building blocks.

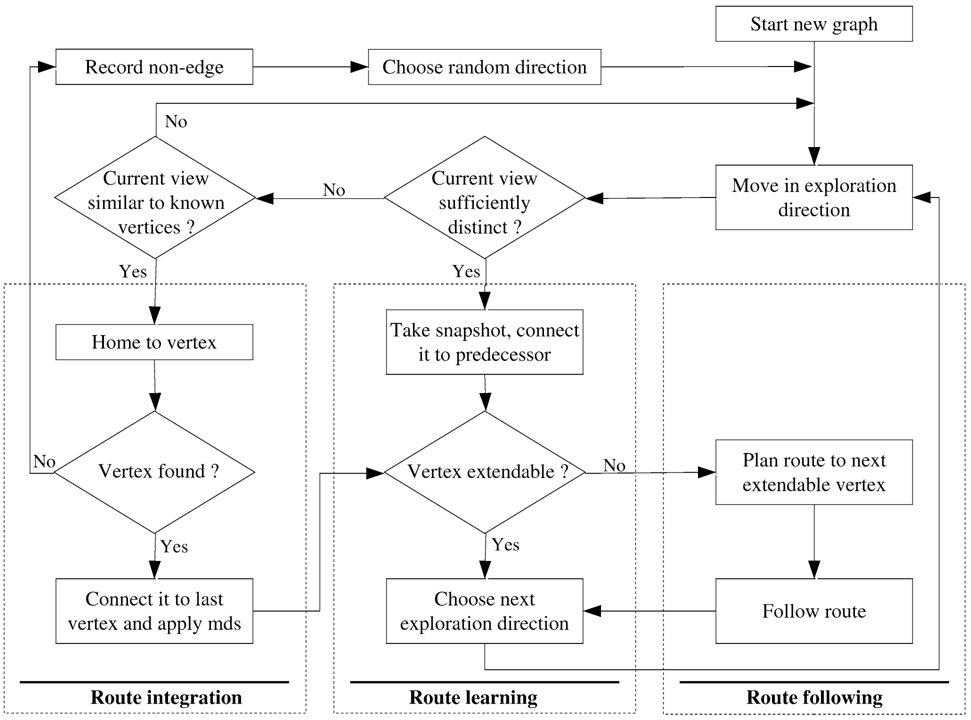
Start new graph
Record non-edge
Choose random direction
No
Current view similar to known vertices ?
No
Current view sufficiently distinct ?
Move in exploration direction
Yes
Yes
Home to vertex
Take snapshot, connect it to predecessor
No
Vertex found ?
Vertex extendable ?
No
Plan route to next extendable vertex
Yes
Yes
Connect it to last vertex and apply mds
Choose next exploration direction
Follow route
Route integration
Route learning
Route following

Fig. 4 Generic exploration scheme. The exploration algorithm has been adapted partially from Franz et al. (1998a). The algorithm consists of three main parts. *Route learning* is responsible for extension of the graph into unknown areas. *Route integration* is responsible for closing large loops and for local mesh refinement. *Route following* is used to coordinate single exploration goals within a local focus region (details are explained in the text)

Route Learning. The route learning module is responsible for extending the map into unknown areas. New vertices should be recorded at locations which are most distinguishable from all other locations. In a small neighborhood, where the uncertainty of pose estimates may not allow a proper disambiguation, distinctiveness is based on the image similarity. At a more global scope, where the spatial aliasing problem becomes important, distinctiveness is guaranteed by global pose estimates as derived in the previous section. Further, suitable node locations are constrained by the path integration noise, i.e. the total number and density of nodes should be high enough to allow regular recalibration of path integration. This becomes crucial if the agent has to navigate in small passages between obstacles, especially because catchment areas are small near obstacles.

For these reasons the average edge length is kept inside an interval $[d_{\min}, d_{\max}]$. A new snapshot is recorded if the estimated distance to the last node is larger than $d_{\min}$ and the image dissimilarity exceeds a certain threshold, i.e. in the interval the location of the snapshot is correlated with the liv-factor. A new snapshot is always recorded if the distance to the last node is larger than $d_{\max}$.

After adding a new node and linking it to its predecessor, the robot rotates about a fixed angle (angle β in Fig. 5(b)) and continues exploration into the new direction. The angle can be used to determine the speed at which the graph is extended into open space. Here we choose a rotation angle of 85° which has turned out to be a good compromise between map extension and map consolidation.

Route Integration. The major purpose of the route integration module is to close loops. Therefore, the instantaneous view is continuously compared to stored views, while the robot explores unknown areas. In order to avoid the extensive search of all stored snapshots, a focus region is defined. The size of the focus region depends on the uncertainty of the path integrator, i.e. it continuously increases while moving in unmapped areas. If the current view becomes similar to a stored view, the robot starts a homing trial. If it succeeds, the target node is linked to the starting node and a new loop is closed. If it fails, because the homing took too many iterations or because the robot was repeatedly forced to avoid obstacles, the edge is marked as not traversable and the robot tries to re-localize itself on the map.

The second purpose of the route integration module is to refine the graph mesh, i.e. to split existing circuits into smaller ones. Increasing the connectivity of vertices increases the amount of certainty of the global pose estimate. The dependence between connectivity and certainty can directly be seen from the gradient of function (10). As known nodes attract the agent while moving between unlinked nodes, the graph usually stays planar. Homing trials within a circuit are not critical, because the robot is surrounded by known places, allowing a fast localization in case of a failure. Therefore, the metric neighborhood used for finding matching snapshots coincides with the graph neighborhood used for localization.

A refinement step is only initiated from extendable nodes, i.e. nodes where the largest open angle between two neighboring edges exceeds a threshold (see Fig. 5), i.e. $\alpha =$

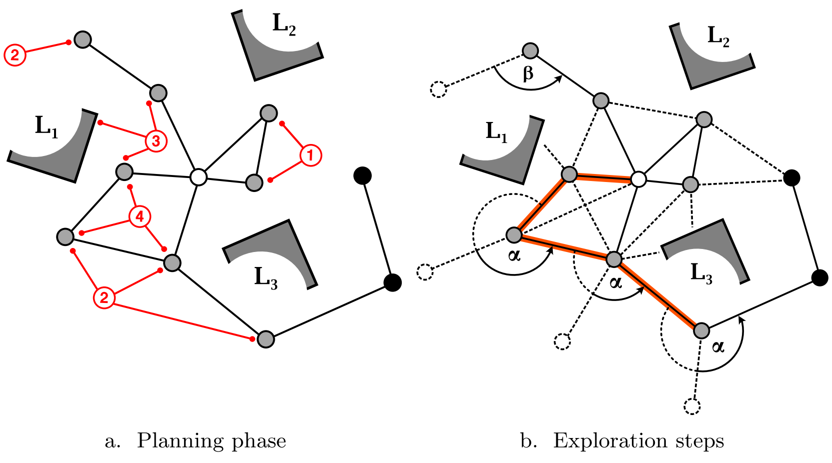

Fig. 5 Exemplary exploration step. In the illustrated example the agent is located at an unextendable node (*white node in the middle*). In this situation the agent plans a route to an extendable node (see Fig. 4). For route planning only nodes close to the current location (*gray dots*) are considered. Each extendable node is associated with a list of possible exploration steps, depicted as *red numbers* in (**a**). The expected outcome of an exploration step is depicted as dashed lines in (**b**). In detail possible exploration steps are: (1) Closing a loop. (2) Extending the graph into unknown areas. (3) Verifying the existence of non-edges and dead-end-edges. (4) Refining the node mesh. In order to determine a target node, one exploration step is selected according to a non-deterministic decision policy. One node that supports the exploration step is chosen at random from the initial node set. Finally a route to the chosen target node is calculated, based on the expected information gain along the route (*bold lines* in (**b**))

$\frac{1}{2}\max\arctan(\frac{y_j-y_i}{x_j-x_i}) \geq$ const. Moving into the largest open angle ensures that the graph is extended equally in all directions. Deadlock situations may occur if the largest open angle guides the robot towards an obstacle. To avoid repeated obstacle avoidance steps, the exploration direction is recorded as a dead-end-edge in case the robot hits an obstacle before recording a new node. Next time the robot tries to extend the same node, moving into the direction of the largest open angle guides the agent away from the old collision course. This guarantees that exploration occurs in all directions, also for nodes located near obstacles.

Route Following. As can be seen from Fig. 4, route following becomes active if the robot is located at a node which cannot be further extended. In this case it becomes necessary to select another extendable node, i.e. to decide which goal to achieve next. In principal it is possible to base the decision of which goal to fulfill next just on local information. This amounts to simply selecting the nearest extendable node. Such a method implies that the agent is not directly forced to extend the map into unknown areas. In general the expected learning progress of such an agent is poor. Therefore, again a focus region (subgraph) for route planning is defined (gray nodes in Fig. 5(b)). Adjusting the size of the focus region allows to define a smooth transition between local map consolidation and expanding the map into unknown areas.

The desired goal is selected randomly from inside the focus region, using a predefined probability distribution. There a two reasons for using a non-deterministic policy. First of all it allows to avoid deadlock situations, which can occur at different time scales and therefore it is difficult to derive an explicit model. Second, it is a simple method to control the overall exploration behavior. Although the policy here uses parameters that have to be adjusted by the programmer it is straightforward to use it as a basis for reinforcement learning methods (e.g. Sutton and Barto 1998; Kaelbling et al. 1996).

Finally a route to the desired goal node is calculated. Path planning is done with the Dijkstra-algorithm (West 1996), which usually uses edge lengths to calculate shortest paths. Here, we use the number of measurements recorded between two nodes as edge weights. This ensures that measurements are equally distributed over the edge set.

5 Results

In a first experiment, we apply the proposed method to solve a trajectory reconstruction problem using the robotic setup shown in Fig. 6. In this case the robot followed a predetermined trajectory, i.e. no exploration was involved. In a second experiment the full system, i.e. mapping and exploration, is tested in a simulation of the robot setup. We use the simulation in order to test the exploration ability in an environment which is much larger than the original arena.

5.1 Trajectory reconstruction

The robotic setup consists of an arena with a size of 140×120 cm^2 (see Fig. 6) containing toyhouses as landmarks.[4]

[4]This setup has already been used in previous work (Franz et al. 1998a, 1998b; Stürzl 2004; Hübner 2005).

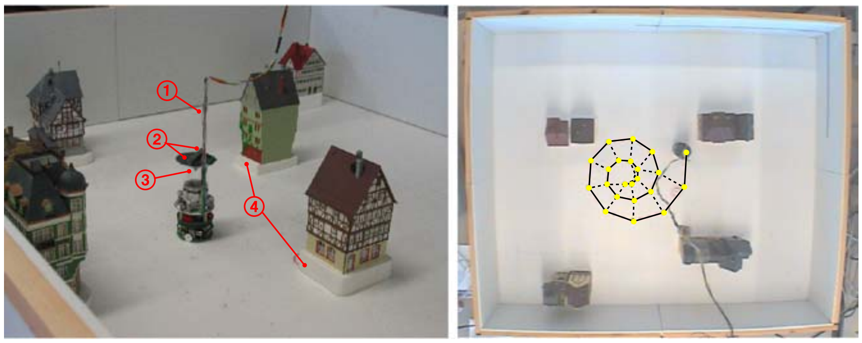

Fig. 6 *Left*: The robotic setup consists of an arena with a size of 140 cm × 120 cm containing toyhouses as landmarks. The toyhouses are placed upon white sockets (4) which can be detected by the IR-sensors more easily. The robot is equipped with two different colored LEDs on its top (2). The LEDs are observed by a calibrated camera system used to track the robots pose. The robots omnidirectional vision system (3) consists of a conical mirror mounted above a camera. The robot is connected with a PC through a cable (1). The cable covers 15° of the panoramic view. *Right*: Starting from the center, the robot moved on a spiral trajectory. At each node (yellow dots) a snapshot was recorded and linked to its predecessor. The task was to reconstruct the trajectory by establishing a pose relation network. Pose relations were determined from odometry (*bold lines*) and from vision (*dashed lines*)

The toyhouses are placed upon white sockets making the obstacle avoidance more reliable. The threshold for the IR-sensor responses has been adjusted so that the *on*-signal corresponds to a distance of approximately 1.5 cm to an obstacle. We use a Khepera miniature table robot. The odometry of the robot has been calibrated in order to remove systematic errors using the method described in Borenstein and Feng (1996). The communication cable covers 15° of the panoramic view. Pixels in this “blind spot” are determined by interpolation. Movements of the robot are evaluated by a tracking system. The tracking system consists of a calibrated camera observing the arena from above (see Fig. 6(b)) and two differently colored LEDs mounted on top of the robot. The tracking system is able to determine the robot’s pose with an accuracy of 2° for heading and 1 cm for position.

The robot traversed a spiral trajectory, recording snapshots and odometry measurements at predefined intervals (see Fig. 6(b)). In order to build a pose relation network, additional links have been added as illustrated by the dashed lines in Fig. 6(b). Derivation of pose relations along these links is described below. From the collected data we reconstructed the recording locations using equation 10. The collected data was evaluated offline under 2 different conditions.[5]

In the first condition the visual home vector (see (1)) was directly used to determine pose relations between adjacent nodes. While the direction (ρ^h) can be estimated quite well, it is not possible to determine the absolute distance (d^h) without knowledge about the landmark distribution. The visual homing method solves this problem by assuming that all landmarks are located on a circle around the current position (“equal-distance-assumption”, Franz et al. 1998b). The radius (R) of the circle is defined as a constant for the whole environment. Hence, distances extracted from two snapshots can be determined up to a factor R: $d^h \propto R \Rightarrow d^h = \lambda R$. The scaling effect can be seen in Fig. 7(a) and Fig. 7(b). In addition to poor scaling, position estimates in Fig. 7(a) are affected by error accumulation. Position estimates in Fig. 7(b) are still poorly scaled, but the shape of the original trajectory is correctly determined.

[5]The data was processed offline in order to ensure that all evaluations are based on the same measurements.

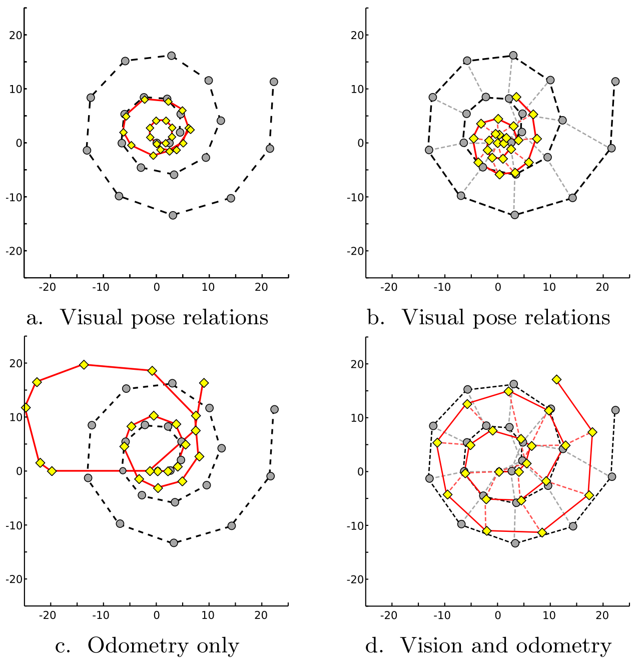

Fig. 7 Trajectory reconstruction task with the 72 pixel panoramic view. True node locations are shown as gray dots, estimated node locations are shown as yellow dots. Distances are measured in cm. **a** and **b** illustrates the case where only visual information is used. **c** and **d** illustrates the case where visual information and measurements from odometry are integrated into one map

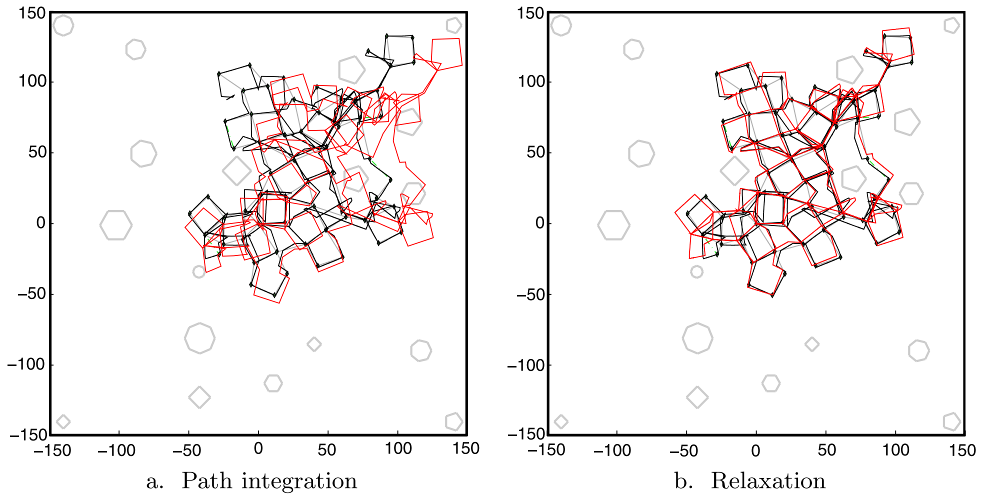

Fig. 8 Effectiveness of the relaxation method. *Black lines* show the true trajectory during exploration. *Red lines* show the estimated trajectory by the agent. *Left*, trajectory is estimated by an uncalibrated path integrator. *Right*, estimated trajectory applying the relaxation algorithm. Simulated arena size is 300 × 300 cm^2

In the second condition global position estimates are derived from pose relations determined by odometry[6] and pose relations determined by the visual homing method at the same time. In order to scale both types of pose relations to the same order of magnitude we adapted the landmark distance locally according to odometry measurements (d^{odo}) between adjacent nodes. Therefore, the distance (d^h_{ab}) between two nodes, a and b, is calculated from the edge set of neighboring nodes (i.e. nodes where odometry measurements have already been recorded) without active homing.

$$d^{\text{odo}} \overset{!}{\approx} d^h = \lambda R \quad \Rightarrow \quad R \approx \frac{d^{\text{odo}}}{\lambda}, \tag{20a}$$

$$d^h_{ab} = \lambda_{ab} \frac{1}{n} \sum_{i=1}^{n} R_i = \lambda_{ab} \left\langle \frac{d_i^{\text{odo}}}{\lambda_i} \right\rangle,$$
$$n = |\{c|(c,a) \in E \vee (c,b) \in E\}|. \tag{20b}$$

The result is shown in Figs. 7(c) and 7(d). Figure 7(c) clearly shows the effect of error accumulation. Figure 7(d) shows the result from the pose relation network when the average landmark distance is locally adapted according to (20b). The shape of the trajectory and the scale are correctly determined after inconsistencies from the pose relation network have been removed.

5.2 Exploring unknown environments

Here we use a simulation of the robotic setup in order to test the exploration ability in environments which are much larger than the original arena. Parameters for the simulated odometry as well as distance and noise characteristic of the simulated IR-sensors have been adapted to the hardware. The panoramic vision system has been substituted by a ray-tracing algorithm. The environment used here is an arena with a size of 9 m^2, surrounded by a white wall. The arena includes randomly placed obstacles of different size, reflectance, and shape.

One of the major applications of the embedded view graph is to provide a reliable method for regularly recalibrating the path integrator. Figures 8(a) and 8(b) show examples of the recalibration performance. The black line is the true trajectory traversed during exploration. Overlayed is the red curve, showing the path integrator state, with optimization—Fig. 8(b)—and without optimization—Fig. 8(a).

In contrast to position estimates of the graph nodes, the trajectory includes all error sources, especially selflocalization errors which may be larger than position errors of the nodes alone. The black trajectory is the true one, the red trajectory is estimated by the agent. The estimated trajectory in Fig. 8(a) is measured by a path integrator without recalibration, while the agent explores the environment using the relaxation method. With path integration alone it is not possible to map such a large portion of the environment.

Heading estimates are achieved using the optimization method described in Sect. 3. Figure 9 shows the real and estimated headings during exploration; the deviation does not exceed 10°. Deviations during constant parts of the red line are a result of drift errors during translations which cannot be compensated by the optimization method. Again, heading estimates are a result of recalibration and optimization.

Most applications of metric knowledge in path planning aim at determining new routes from memory, i.e. to find shortcuts over unexplored areas. A straight forward calculation of shortcuts over the graph mesh is problematic. First, because according to the motion model, the probability of a failure increases with the length of the shortcut. Second, because the graph does not contain explicit information about the location of obstacles.

[6]Since the Khepera odometry worked quite well we added an additional noise factor of 10% to odometry measurements.

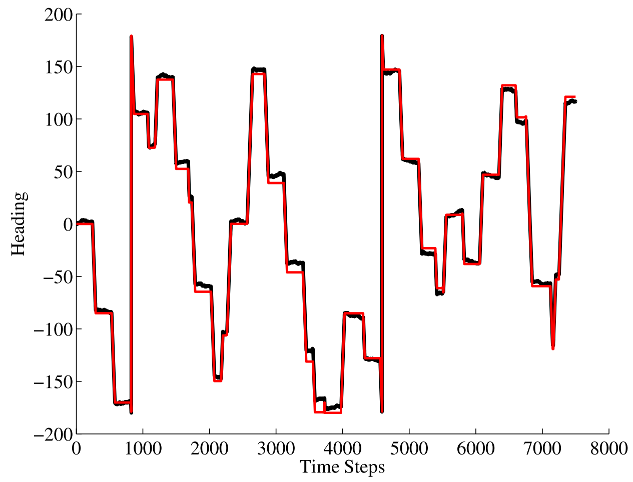

Fig. 9 Heading estimates during exploration. This plot shows recalibration of the heading estimation while moving along the trajectory shown in Fig. 6(b). The gray curve is the real heading. The red curve shows the estimated heading applying the optimization method described in (18). Deviations during constant parts of the *red line* result from drift errors which can not be compensated by the optimization method

To address these problems we convert the original graph (Fig. 10(a)) into a graph which allows derivation of efficient and secure routes. As a basis we use the graph consisting of all nodes having a dead-end-edge, i.e. the graph includes all nodes which are assumed to be near obstacles (Fig. 10(b)). These nodes are important, because navigation in narrow passages requires more frequent recalibration than in open space. To this clustered subgraph, all first neighbors of the obstacle nodes are added. Finally, starting from the first node of the route we add each node within the maximum allowed distance for a shortcut and link these nodes to their nearest neighbor, while neighborhood is now measured by the Euclidean distance. Figure 10(c) shows the result of this procedure, using the center node as the start of derived routes.

In order to test the shortcut ability, the agent's task was to follow routes from the center node to a set of randomly selected target locations. The resulting trajectories are illustrated in Fig. 10(c). Figures 11(a) and 11(b) show two of these trajectories in more detail, illustrating course correction and replanning capabilities. The blue nodes in Fig. 11(a) show the planned path. Deviations from the route occur for

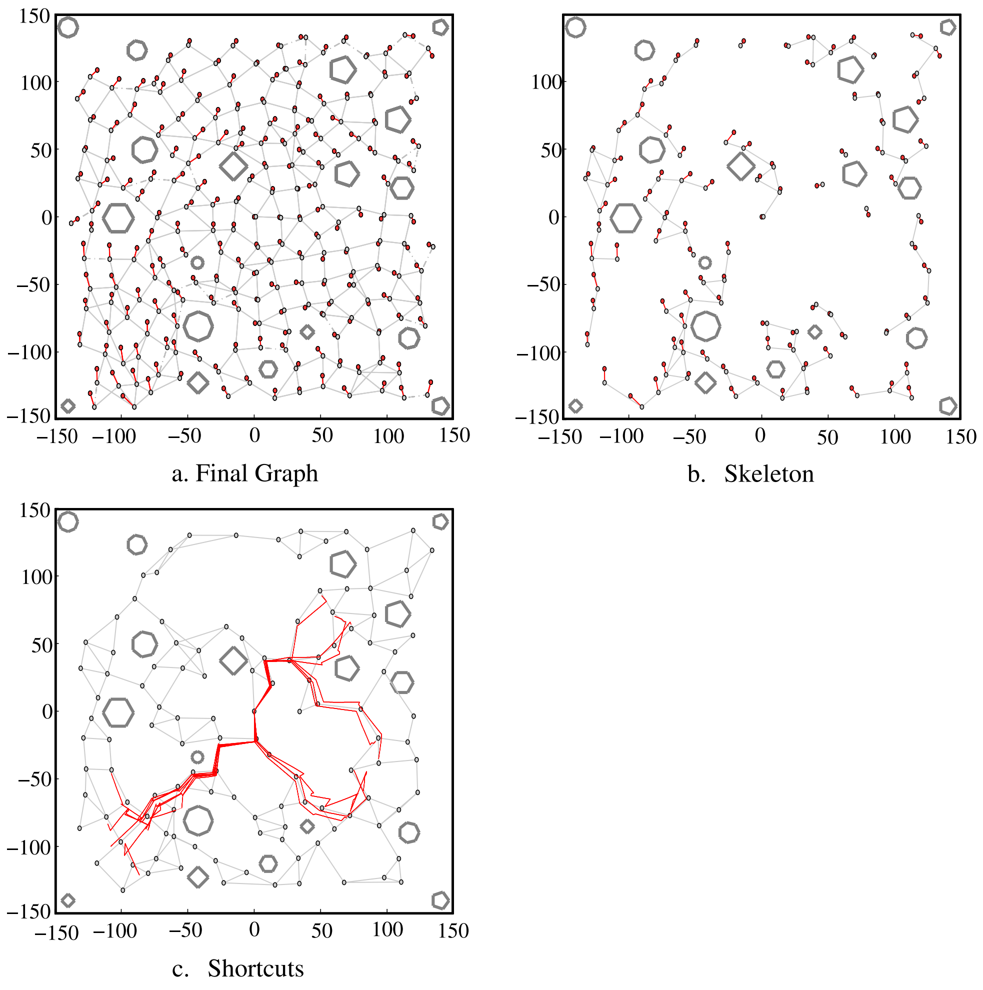

Fig. 10 Exploration results. **a** The final graph covers an environment of 300 × 300 cm^2 with a total of 191 nodes. *Gray dots* indicate the estimated node positions. *Red dots* indicate the true recording locations. **b** Subgraph (skeleton graph) consisting only of nodes which are assumed to be near obstacles. The original graph is split into several unconnected clusters, concentrated around obstacles. **c** Route following including shortcuts. It can be seen that the agent corrects errors during route following. Further, small path ways can be traversed due to the increased node density near obstacles. Routes do not end precisely at the target node because of the region of uncertainty limiting the accuracy of the scene based homing method

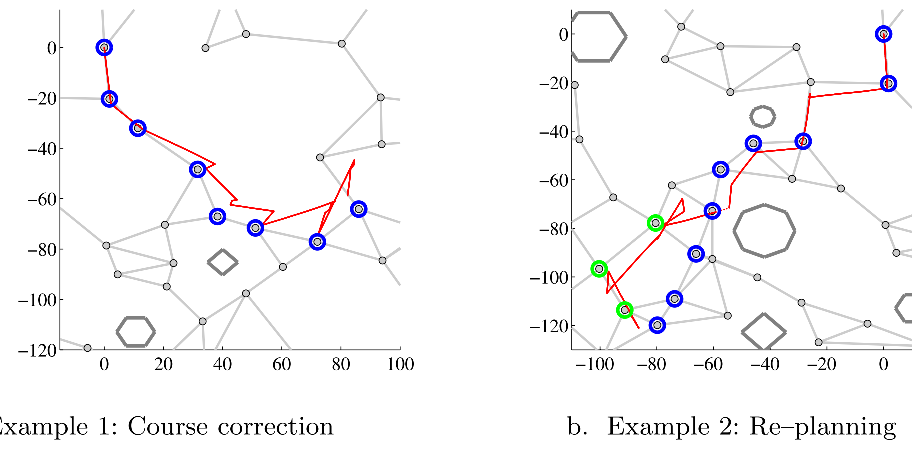

Fig. 11 Example trajectories. These figures show two example trajectories taken from Fig. 10(c). *Blue dots* indicate the original planned route. **a** This example illustrates the ability of course corrections by scene based homing. **b** This example shows the ability to replan routes (*green nodes*) after active obstacle avoidance

three reasons. First, due to odometry errors the agent is not able to follow precisely a calculated path. Second, the global pose estimates are still erroneous (see Fig. 10(a)). Third, the scene based homing algorithm has a limited spatial resolution (region of uncertainty). Therefore, the path integrator is not accurately recalibrated at intermediate vertices and furthermore the trajectory does not end precisely at the desired goal location.

Figure 11(b) shows a second example where the agent tries to move through a small pathway and hits an obstacle. After avoiding the obstacle (dotted part of the trajectory) the agent relocalizes on the map and calculates an alternative route (green nodes) to the target location.

6 Summary

We present a method to learn globally consistent maps based only on weak metric information derived from odometry. The complex navigation task of mapping unknown environments has been based on three local navigation strategies, path integration, scene based homing, and obstacle avoidance.

Alternative to a pose-based control scheme, we applied a homing algorithm, which makes use of visual landmark information and pose information. The combined homing scheme fulfills several purposes. First, it allows automated adaptation to the environment. Second, it has been used to compensate path integration errors, and therefore allows the mapping algorithm to be based solely on weak odometry information. And finally it has been used to decouple metric information from the mapping task, which offers a great flexibility in the choice of the landmark model, ranging from a pure vision based system up to a system capable to achieve survey knowledge.

References

Argyros, A. A., Bekris, K. E., & Orphanoudakis, S. C. (2001). Robot homing based on corner tracking in a sequence of a panoramic images. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR'01)* (Vol. 2).

Borenstein, J., & Feng, L. (1996). Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, *12*(6), 869–880.

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling*. New York: Springer.

Braitenberg, V. (1984). *Vehicles. Experiments in synthetic psychology*. Cambridge: MIT.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, *2*(1), 1423.

Cartwright, B., & Collett, T. (1983). Landmark learning in bees. *Journal of Comparative Physiology A*, *151*, 521–543.

Cassinis, R., Duina, D., Inelli, S., & Rizzi, A. (2002). Unsupervised matching of visual landmarks for robotic homing using Fourier Mellin transform. *Robotics and Autonomous Systems*, *40*, 131–138.

Duckett, T., Marsland, S., & Shapiro, J. (2002). Fast, on-line learning of globally consistent maps. *Autonomous Robots*, *12*(3), 287–300.

Floreano, D., & Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics*, *26*, 396–407.

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

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

Frese, U., Larsson, P., & Duckett, T. (2004). A multigrid algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, *21*(2), 1–12.

Golfarelli, M., Maio, D., & Rizzi, S. (1998). Elastic correction of dead-reckoning errors in map building. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS'98)* (pp. 905–911), Victoria, Canada.

Golledge, R., & Stimson, R. (1997). *Spatial behavior: a geographic perspective*. San Francisco: Guilford.

Gourichon, S., Meyer, J.-A., & Pirim, P. (2002). Using coloured snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation, Special issue on Biologically Inspired Robotics*, *17*(4), 154–162.

Gutmann, J.-S., & Konolige, K. (1999). Incremental mapping of large cyclic environments. In *Proceedings of the IEEE international* *symposium on computational intelligence in robotics and automation (CIRA)* (pp. 318–325), Monterey, California.

Hafner, V., & Möller, R. (2001). Learning of visual navigation strategies. In M. Quoy, P. Gaussier, & J. Wyatt (Eds.), *Proceedings of the European workshop on learning robots (EWLR-9)* (pp. 47–56), Prague.

Hübner, W. (2005). *From homing behavior to cognitive mapping, integration of egocentric pose relations and allocentric Landmark information in a graph model*. PhD thesis, Universität Bremen.

Hutchinson, S., Hager, G., & Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, *12*(5), 651–670.

Jähne, B. (Ed.). (1999). *Handbook of computer vision and applications*. New York: Academic Press.

Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kuipers, B. (2000). The spatial semantic hierarchy. *Artificial Intelligence*, *119*, 191–233.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., & Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems, Special issue on Biomimetic Robots*, *30*, 39–64.

Liu, Y., & Thrun, S. (2003). Results for outdoor-SLAM using sparse extended information filters. In *IEEE international conference on robotics and automation (ICRA)*.

Lu, F., & Milios, E. (1994). Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, *18*, 249–275.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, *4*, 333–349.

MacKay, D. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*, 590–604.

Mallot, H. (1999). Spatial cognition: behavioral competences, neural mechanisms, and evolutionary scaling. *Kognitionswissenschaft*, *8*, 40–48.

Mardia, K., Kent, J., & Bibby, J. (1982). *Multivariate analysis*. New York: Academic Press.

Menegatti, E., Zoccaratoa, M., Pagelloa, E., & Ishiguroc, H. (2004). Image-based Monte Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, *48*, 17–30.

Möller, R. (2000). Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics, Special issue: Navigation in Biological and Artificial Systems*, *83*(3), 231–243.

Möller, R., Lambrinos, D., Roggendorf, T., Pfeifer, R., & Wehner, R. (2001). Insect strategies of visual homing in mobile robots. In B. Webb, & T. Consi (Eds.), *Biorobotics—methods and applications*. Cambridge: AAAI/MIT.

Redish, A. (1999). *Beyond the cognitive map*. Cambridge: MIT.

Rizzi, A., Duina, D., Inelli, S., & Cassinis, R. (2001). A novel visual landmark matching for a biologically inspired homing. *Pattern Recognition Letters*, *22*, 1371–1378.

Röfer, T. (1997). *Controlling a wheelchair with image-based homing*, Technical report, AISB Workshop on spatial reasoning in mobile robots and animals. Tech. rep. UMCS-97-4-1, Dept. Computer Sc., Manchester University.

Se, S., Lowe, D., & Little, J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, *21*(3), 364–375.

Sim, R., & Dudek, G. (1998). Mobile robot localization from learned landmarks. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS'98)* (pp. 1060–1065), Victoria, Canada.

Stürzl, W. (2004). *Sensorik und Bildverarbeitung für Landmarkenbasierte Navigation*. PhD thesis, Universität Tübingen.

Stürzl, W., & Mallot, H. (2002). Vision-based homing with a panoramic stereo sensor. In *Lecture notes in computer science* (Vol. 2525, pp. 620–628).

Sutton, R., & Barto, A. (1998). *Reinforcement learning—an introduction*. Cambridge: MIT.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge: MIT.

Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, *128*(1–2), 99–141.

Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., & Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, *23*(7–8), 693–716.

Vardy, A., & Möller, R. (2005). Biologically plausible visual homing methods based on optical flow techniques. *Connection Science, Special issue: Navigation*, *17*(1–2), 47–89.

Vardy, A., & Oppacher, F. (2003). Low-level visual homing. In W. Banzhaf, T. Christaller, P. Dittrich, J. Kim, & J. Ziegler (Eds.), *Advances in artificial life, proceedings of the 7th European conference on artificial life (ECAL)* (Vol. 2801, pp. 875–884).

Vardy, A., & Oppacher, F. (2004). A scale invariant neural feature detector for visual homing. In G. Palm & S. Wermter (Eds.), *Proceedings of the workshop on neurobotics, German conference on artificial intelligence*.

Weber, K., Venkatesh, S., & Srinivasan, M. (1999). Insect-inspired robotic homing. *Adaptive Behavior*, *7*(1), 65–97.

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

West, D. (1996). *Introduction to graph theory*. New York: Prentice–Hall.

Whaite, P., & Ferrie, F. (1997). Autonomous exploration: driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 193–205.