Auton Robot (2007) 23: 231–245
DOI 10.1007/s10514-007-9043-x

Visual homing in environments with anisotropic landmark distribution

Ralf Möller · Andrew Vardy · Sven Kreft · Sebastian Ruwisch

Received: 21 February 2007 / Accepted: 15 May 2007 / Published online: 19 July 2007

Abstract Gradient descent in image distances can lead a navigating agent to the goal location, but in environments with an anisotropic distribution of landmarks, gradient home vectors deviate from the true home direction. These deviations can be reduced by applying Newton's method to matched-filter descent in image distances (MFDID). Based on several image databases we demonstrate that the home vectors produced by the Newton-based MFDID method are usually closer to the true home direction than those obtained from the original MFDID method. The greater accuracy of Newton-MFDID home vectors in the vicinity of the goal location would allow a navigating agent to approach the goal on straighter trajectories, improve the results of triangulation procedures, and enhance a robot's ability to detect its arrival at a goal.

Keywords Visual homing · Newton's method · Matched filter · Descent in images distances

1 Introduction

For most modern approaches to robot navigation the ability of a robot to reach a goal position is contingent upon the development of a map, wherein both the current and goal positions are represented. These approaches usually exploit SLAM (Simultaneous Localization and Mapping) to develop maps which incorporate numerous real-world features into a 2D or 3D volumetric representation. Such representations can be quite costly to build, use, and update. An alternative approach is to develop a topological representation, wherein the nodes of a graph represent positions where sensory information has been captured, and edges represent adjacency or reachability relations. Franz et al. (1998) suggested that such a topological map could be entirely visual, with *snapshot images* stored at nodes and a *visual homing* procedure used to move the robot from node to node. This form of representation is lightweight—it requires only a sparse sampling of the environment. Further, visual homing obviates the need to store the geometrical relationships between nodes (which can easily become corrupted by noise).

Visual homing methods compute the home direction from a stored snapshot image and the image seen at the current location (reviews: Möller and Vardy 2006; Vardy and Möller 2005; Zeil et al. 2003; Franz and Mallot 2000). The typical solution to this particular problem of ego-motion estimation is to establish correspondences between local regions in the two images, for example by optical flow methods (Vardy and Möller 2005), transform the correspondence vectors into movement vectors, and average the movement vectors to determine the home direction.

A fresh perspective was brought to this field by the observation of Zeil et al. (2003) that distance measures between compass-aligned panoramic images usually increase smoothly with spatial distance between the corresponding camera positions. Thus, any movement strategy that reduces the image distance will also reduce the spatial distance from the goal, therefore the term *descent in images distances (DID)* can be used for this class of methods. The DID approach transforms the visual homing task from a correspondence problem into an optimization problem, so that novel

R. Möller (✉) · S. Kreft · S. Ruwisch
Computer Engineering, Faculty of Technology, Bielefeld University, 33594 Bielefeld, Germany

A. Vardy
Computer Science/Engineering & Applied Science, Memorial University of Newfoundland, St. John's, A1B 3X5, Canada

solutions can be derived by resorting to the arsenal of optimization methods; Zeil et al. (2003) applied simple numerical gradient descent methods (versions of the Gauss–Seidel strategy and of Evolutionary Operation). Recently, Zampoglou et al. (2006) described a DID method inspired by models of chemo-taxis.

Three significant problems remain with the DID approach to visual homing:

1. The estimation of the gradient of the image distance function requires sampling images from adjacent positions within the environment. This sampling requirement introduces additional complexity and cost.
2. In environments with an anisotropic distribution of landmarks the image distance function becomes distorted from the ideal case of pure radial symmetry.
3. DID cannot proceed to estimate the direction of translation unless the images are aligned to a common compass direction. This requires some form of orientation estimator.

The first point has already been addressed in our previous paper (Möller and Vardy 2006). In that work we developed a technique to approximate the gradient of the image distance function without the need to sample adjacent positions within the environment. The second point is tackled in this paper and discussed in more detail below. As for the third point, we utilize the DID compass method suggested by Zeil et al. (2003). We test for the impact of errors of the DID compass in our experiments.

To illustrate the second point, Fig. 1 visualizes the Euclidean distance for two virtual rooms with two or four textured walls, respectively (see also Fig. 5). The distance was computed between a snapshot image taken at the position marked with the cross and 19 × 19 images on a grid covering the room. It is apparent that in a wide region around the snapshot position, a gradient descent in the image distance function would lead an agent back to this position. However, when landmark features are anisotropically distributed in the environment, the Euclidean distance rises with different slope in different directions from the snapshot position. In the virtual room with two opposite walls (Fig. 1, top), the image distance changes fast when the agent moves parallel to the walls, and more slowly when it moves perpendicular to the walls, resulting in elongated iso-distance curves. In contrast, with a roughly isotropic landmark distribution as in the four-walled room (bottom), iso-distance curves assume a more circular shape. Note that the particular shape of the image distance function depends also on the specific visual features that the environment presents.

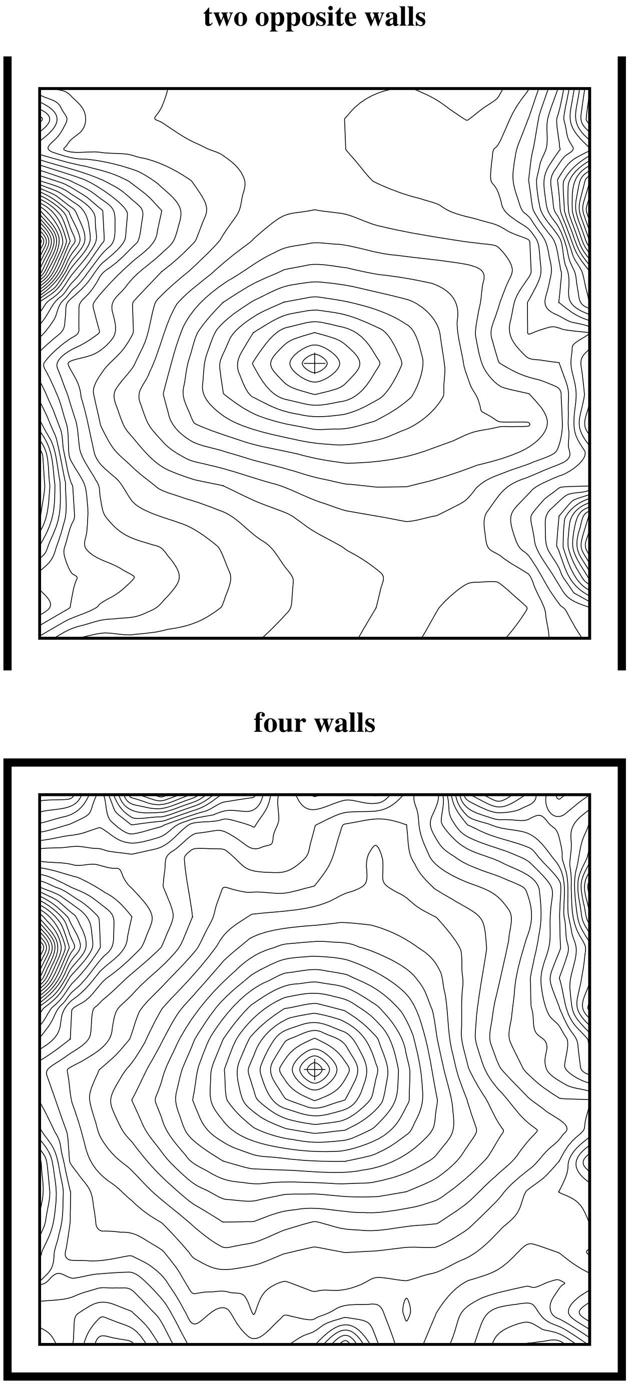


Fig. 1 Euclidean distance in two virtual rooms between a snapshot image taken at the position marked with a cross and 19 × 19 images on a grid within the inner square. *Top*: Virtual room with two textured walls (*thick lines*) and an otherwise uniform background, *bottom*: four textured walls. Cutoff frequency: 0.05

A navigating agent following the gradient of the image distance in an anisotropic environment may therefore approach the snapshot location on a curved path which, besides being less efficient, may not be desirable for certain applications (e.g. cleaning robots). Moreover, since some of the gradient home vectors are not pointing towards the snapshot location, they cannot be used for purposes of deriving metrical information from visual cues, like triangulation or retroactive annotation of topological maps with met-

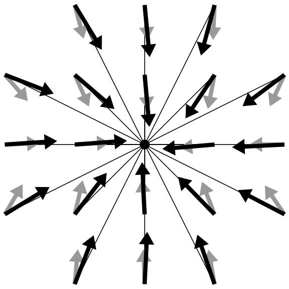


Fig. 2 Gradient home vectors (*gray*, *short*) and Newton home vectors (*black*, *long*) obtained from the MFDID method in the virtual room with two opposite textured walls. The snapshot position (*dot*) coincides with the snapshot position in Fig. 1 (*top*). The home vectors are computed for 20 neighboring positions on the grid. Thin lines indicate the true home direction. Cutoff frequency: 0.05

rical information (Hafner 2000; Hübner and Mallot 2002; Hübner 2005). These deviations between gradient home vectors and true home directions are clearly visible in the gray vectors in Fig. 2 obtained for the virtual room with two walls; the home vectors were determined by matched filter descent in image distances (MFDID, Möller and Vardy 2006). A similar effect was observed when MFDID was applied to virtual rooms with sparsely distributed features (Vardy 2006, Fig. 5).

The deviation of gradient vectors from the true direction to the minimum is a common problem in optimization. Newton's method and its numerous offshoots can correct for these deviations, at least in the vicinity of the minimum point (see e.g. Chong and Żak 2001); here we use Newton's method since we strive for a closed-form solution rather than an iterative method. In our previous work (Möller and Vardy 2006), we already derived a closed-form solution for the gradient descent in image distances based on the matched filter concept (*matched-filter descent in image distances, MFDID*). Two translational optical flow fields—the matched filters—are used to predict changes of the image distance function, so that the gradient home vector can directly be computed without the necessity to sample the image distance function at different locations (see Fig. 3). MFDID is based on the assumption that all landmark features are located at approximately the same distance from the vantage point.

The gradient equation of the MFDID method is the starting point for the derivation of a *Newton-based version of matched-filter descent in image distances*, in the following called *Newton-MFDID*. We derive a closed-form solution for Newton-MFDID based on a modified Newton method where the Hessian is determined at the goal location. This is a particular characteristic of applying the Newton method to the visual homing problem: While in a standard optimization problem the goal location (optimum) is not known but has to be determined and thus the Hessian for this point is also not available, the goal location has previously been visited in the visual homing problem and the Hessian can therefore be determined for this location. This has implications for the range of applicability of the Newton descent, since the Hessian remains unchanged and, in contrast to the standard Newton method, cannot lose its positive definiteness at larger distances from the goal. As the gradient-based MFDID method, Newton-MFDID rests on an equal-distance assumption as well as some Taylor approximations and therefore only delivers an *approximation* of the gradient and Hessian. It is not clear whether under these conditions the Newton method will actually improve the home vector quality; only tests on image data (real images and data from virtual environments) can provide insights on the applicability of the method. Here we compare the performance of MFDID and Newton-MFDID on six different image databases.

After a summary of the notation (Sect. 2), we present a short description of the gradient-based DID method and its Newton version (Sects. 3 and 4). In *DID* and *Newton-DID*, gradient and Hessian are directly determined from multiple images; in our case, these images are taken from a database. We then recapitulate the essential equations of *MFDID* in Sect. 5 and derive the novel *Newton-MFDID* method in Sect. 6. Section 7 briefly explains the DID compass which can be used to align snapshot and current view. The performance of the four methods (DID, Newton-DID, MFDID, Newton-MFDID), expressed by the angular error between home vectors and true home direction, is analyzed in Sect. 8 for six image databases, four of them collections of real-world images, two of them obtained from image rendering. There we also analyze how the performance of MFDID and Newton-MFDID is affected by errors in the DID compass. As will be shown in Sect. 9, Newton-MFDID is significantly superior to the original, gradient-based MFDID method in five of the databases, and not significantly worse in the remaining one. We can also demonstrate that MFDID and Newton-MFDID are sufficiently robust against errors in the relative orientation of the images when used together with the DID compass.

2 Notation

We use the same notation as in our previous work (Möller and Vardy 2006): Let $\boldsymbol{\varphi}_{ij} = (\beta_i, \gamma_j)^T$ be a two-dimensional vector of angles in spherical coordinates describing the direction from the vantage point of the camera to the visual feature in the environment which appears at pixel coordinates $(i, j)^T$ in the image. The horizontal angle β_i (azimuth) varies only with pixel coordinate i, the vertical angle γ_j (elevation) only with j ($\gamma_j = 0$ corresponds to the horizon, positive γ_j are above the horizon). In the horizontal direction, the image is closed to a panoramic view. Also, let $\mathbf{x} = (x, y)^T$ be the position of the vantage point in the horizontal plane. We assume that the camera coordinate system is aligned with a fixed world coordinate system by using some compass. With $C(\boldsymbol{\varphi}_{ij}, \mathbf{x})$ we describe a pixel $(i, j)^T$

in the image taken at robot position $\mathbf{x}$. In local visual homing, a snapshot image is captured at the goal location $\mathbf{x}_0$ and stored, in the following denoted as $S(\boldsymbol{\varphi}_{ij}) = C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0)$. All images are monochrome. A grid position (m, n) in an image database corresponds to a spatial position $\mathbf{x} = (x_m, y_n)$.

3 Gradient-based DID

In all four methods presented in this paper, the image distance is computed from the sum of squared errors (SSE) between the current image $C(\boldsymbol{\varphi}_{ij}, \mathbf{x})$ and the snapshot $S(\boldsymbol{\varphi}_{ij})$:

$$p(\mathbf{x}) = \frac{1}{2} \sum_{i,j} [C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) - S(\boldsymbol{\varphi}_{ij})]^2. \tag{1}$$

In the following, we also use the term *potential* for such a distance measure. The home vector $\mathbf{h}_g(\mathbf{x})$ in gradient-based DID coincides with the spatial gradient in the image distance $p(\mathbf{x})$. In a moving robot, the gradient could be estimated by inserting perpendicular test steps into the movement and sampling the image distance from (1) at *three* non-collinear points. In our database experiments, we instead compute the gradient at a grid position (m, n) by approximating the first-order derivatives from the images at *four* neighboring grid positions for reasons of symmetry:

$$\begin{aligned} \mathbf{h}_g(\mathbf{x}) &= \mathbf{h}_g([x_m, y_n]) = -\nabla_{\mathbf{x}} p(\mathbf{x}) \\ &\approx \frac{1}{2} \begin{pmatrix} p([x_{m+1}, y_n]) - p([x_{m-1}, y_n]) \\ p([x_m, y_{n+1}]) - p([x_m, y_{n-1}]) \end{pmatrix}. \end{aligned} \tag{2}$$

Note that for this computation we exclude a margin of one from the database grids in our experiments (Sect. 8).

4 Newton-based DID

In Newton descent, the gradient $\nabla_{\mathbf{x}} p(\mathbf{x})$ at the current position $\mathbf{x}$ is multiplied by the inverse Hessian at this position $\mathbf{H}(\mathbf{x})^{-1}$. The Hessian describes the shape of the distance function, and the inverse Hessian modifies the direction of the gradient such that it points towards the goal location $\mathbf{x}_0$ (see e.g. Chong and Żak 2001). However, while in optimization problems the minimum position $\mathbf{x}_0$ is unknown and therefore the Hessian $\mathbf{H}(\mathbf{x})$ has to be determined at the current position $\mathbf{x}$, the image at the goal position $\mathbf{x}_0$ is available in our case and thus the Hessian $\mathbf{H}(\mathbf{x}_0)$ can be computed instead (this version of Newton's method can be derived by a Taylor expansion of $p(\mathbf{x})$ at $\mathbf{x}_0$, see Appendix 1). This has two implications for the homing process: First, the inverse Hessian only has to be determined once. In Newton-DID, the required samples in the vicinity of the goal location could be collected on short-distance exploratory excursions. Second, the Hessian cannot lose its positive definiteness. In standard optimization problems, this property will usually be lost in larger distance from the goal location and the Newton descent will fail, but since in the navigation context we only use the Hessian at the goal, this problem only occurs in rare cases.

The home vector of Newton-DID is determined from

$$\mathbf{h}_n(\mathbf{x}) = -\mathbf{H}(\mathbf{x}_0)^{-1} \nabla_{\mathbf{x}} p(\mathbf{x}) \approx \hat{\mathbf{H}}(\mathbf{x}_0)^{-1} \mathbf{h}_g(\mathbf{x}), \tag{3}$$

where $\hat{\mathbf{H}}(\mathbf{x}_0)$ is a discrete approximation of the Hessian (see below) and $\mathbf{h}_g(\mathbf{x})$ is obtained from (2). In our database experiments, the components of the approximated Hessian are obtained from the 3×3 neighborhood in the grid:

$$\begin{aligned} \hat{H}_{xx}([x_m, y_n]) &= p_{m-1,n} - 2p_{m,n} + p_{m+1,n}, \\ \hat{H}_{xy}([x_m, y_n]) &= (p_{m-1,n-1} + p_{m+1,n+1} \\ &\qquad - p_{m-1,n+1} - p_{m+1,n-1})/4, \\ \hat{H}_{yy}([x_m, y_n]) &= p_{m,n-1} - 2p_{m,n} + p_{m,n+1}, \end{aligned}$$

where $p_{i,j} = p([x_i, y_j])$ and $\hat{H}_{yx} = \hat{H}_{xy}$. As for the computation of the gradient in (2), we leave out a margin of one from the grids. To ensure strictly positive definiteness (hence invertibility), we have to test that $\hat{H}_{xx}(\mathbf{x}_0) > 0$ and $\det \hat{\mathbf{H}}(\mathbf{x}_0) > 0$; in our simulations, these conditions were usually fulfilled, except in a small fraction of snapshot positions for extremely lowpass-filtered images (cutoff 0.01) in the two virtual room databases.

5 Gradient-based matched-filter DID

Matched-filter DID (MFDID) was introduced in our previous work (Möller and Vardy 2006). Figure 3 explains the idea of the method. For the estimation of the gradient in the potential, three images from non-collinear positions in space are required. In the standard DID method and its Newton equivalent described in Sects. 3 and 4, the three samples of the distance function have to be collected by inserting test steps into the movement. In MFDID, these tests steps are avoided by computing two of the images from the current image in a prediction process. The prediction is based on two "matched filters", flow templates for translational movements in perpendicular directions. Note that in the MFDID method in (6), the image prediction is only contained implicitly. The prediction rests on the assumption that the distances to all features in the current view are the same. This assumption, while generally untrue, nevertheless allows sufficiently accurate prediction for the purposes of incremental visual homing Möller and Vardy (2006). Binding and Labrosse (2006) have independently developed a visual navigation method which shares the core idea of MFDID that

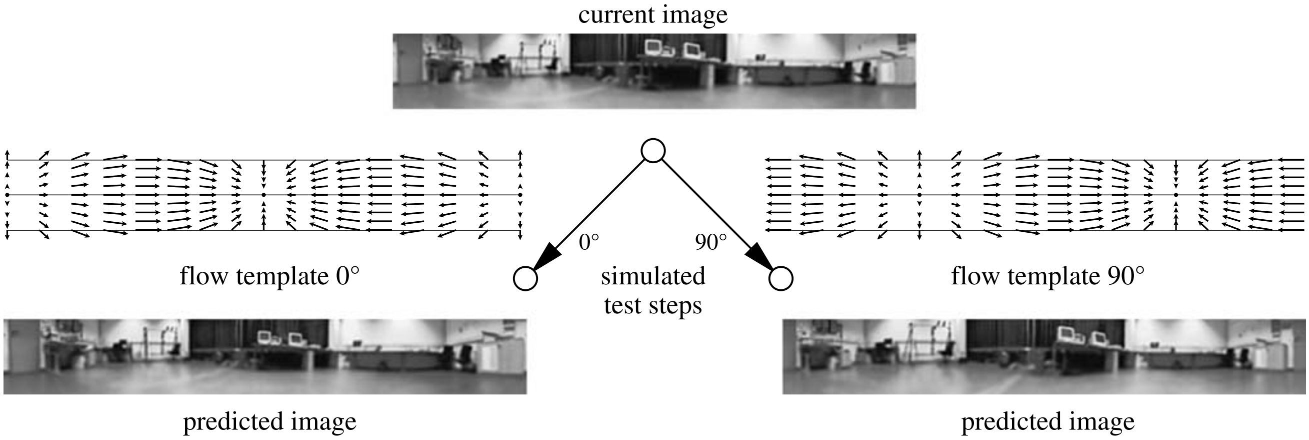


Fig. 3 Matched-filter descent in image distances. Two images are predicted by applying two translational flow templates for perpendicular movement directions to the current image. (For the figure, images from the database were taken instead of predicted images)

two images are predicted from the current view. However, in their approach, the image prediction is done *explicitly* by an approximative "image warping" procedure.

In the MFDID method, the home vector $\mathbf{h}_g^*(\mathbf{x})$ is the negative spatial gradient of the potential defined in (1),

$$\mathbf{h}_g^*(\mathbf{x}) = -\nabla_{\mathbf{x}} p(\mathbf{x}), \tag{4}$$

which gives

$$\mathbf{h}_g^*(\mathbf{x}) = -\sum_{i,j} \nabla_{\mathbf{x}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) \cdot [C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) - S(\boldsymbol{\varphi}_{ij})].$$

The computation of the home vector requires an estimate of the spatial gradient $\nabla_{\mathbf{x}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x})$. As we have shown in our previous work (Möller and Vardy 2006), the spatial gradient can be expressed as a function of the image gradient. This relation was derived by combining two equations. The first is an expression for the optical flow $\Delta\boldsymbol{\varphi}_{ij}$ under pure translations $\Delta\mathbf{x}$ in the plane,

$$\Delta\boldsymbol{\varphi}_{ij} = \frac{1}{D_{ij}} \boldsymbol{\Gamma}(\gamma_j)\mathbf{B}(\beta_i)\Delta\mathbf{x},$$

where D_{ij} is the distance to the feature visible at pixel $(i, j)^T$ and

$$\boldsymbol{\Gamma}(\gamma) = \begin{pmatrix} \sec\gamma & 0 \\ 0 & \sin\gamma \end{pmatrix}, \qquad \mathbf{B}(\beta) = \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix}.$$

Two flow fields $\Delta\boldsymbol{\varphi}_{ij}$ for perpendicular movements $\Delta\mathbf{x}$ in the plane are shown in Fig. 3. The second equation is obtained from a Taylor expansion of $C(\boldsymbol{\varphi}, \mathbf{x})$ (neglecting quadratic and higher-order terms) combined with the assumption that the intensity of a shifted pixel remains constant. This leads to an estimate of the spatial gradient for pixel $(i, j)^T$ expressed as a function of the image gradient $\nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x})$:

$$\nabla_{\mathbf{x}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) \approx -\frac{1}{D_{ij}} \mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}). \tag{5}$$

Under the assumption that all features have the same distance D from the vantage point, we get an expression for the gradient home vector

$$\mathbf{h}_g^*(\mathbf{x}) \approx \sum_{i,j} \frac{1}{D} \mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) \cdot [C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) - S(\boldsymbol{\varphi}_{ij})]. \tag{6}$$

The home vector depends on the first-order image derivatives in the current view and on the difference image between current view and snapshot.

6 Newton-based matched-filter DID

Instead of a gradient descent in the potential from (4), we now apply a Newton descent

$$\mathbf{h}_n^*(\mathbf{x}) = -\mathbf{H}(\mathbf{x}_0)^{-1} \nabla_{\mathbf{x}} p(\mathbf{x}) \approx \mathbf{H}(\mathbf{x}_0)^{-1} \mathbf{h}_g^*(\mathbf{x}), \tag{7}$$

where $\mathbf{H}(\mathbf{x})$ is the Hessian of $p(\mathbf{x})$

$$\mathbf{H}(\mathbf{x}) = \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T p(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 p}{\partial x^2} & \frac{\partial^2 p}{\partial x \partial y} \\ \frac{\partial^2 p}{\partial y \partial x} & \frac{\partial^2 p}{\partial y^2} \end{pmatrix}$$

and $\mathbf{h}_g^*(\mathbf{x})$ is the gradient home vector from (6). Again, as described in Sect. 4, the image at the goal position $\mathbf{x}_0$ is available in our case and thus the Hessian $\mathbf{H}(\mathbf{x}_0)$ can be used. We proceed by computing the derivatives of the gradient $\nabla_{\mathbf{x}} p(\mathbf{x})$ from (4) and (6) and obtain

$$\mathbf{H}(\mathbf{x}) = -\sum_{i,j} \frac{1}{D} \mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \{\nabla_{\boldsymbol{\varphi}} \nabla_{\mathbf{x}}^T C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) \cdot [C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) - S(\boldsymbol{\varphi}_{ij})] + \nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}) \nabla_{\mathbf{x}}^T C(\boldsymbol{\varphi}_{ij}, \mathbf{x})\}.$$

Since $C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0) = S(\boldsymbol{\varphi}_{ij})$, the Hessian at $\mathbf{x}_0$ is

$$\mathbf{H}(\mathbf{x}_0) = -\sum_{i,j} \frac{1}{D} \mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \cdot \nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0) \nabla_{\mathbf{x}}^T C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0).$$

Now we need an expression for $\nabla_{\mathbf{x}}^T C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0) = \nabla_{\mathbf{x}}^T S(\boldsymbol{\varphi}_{ij})$. This can directly be obtained from (5), and with $\nabla_{\boldsymbol{\varphi}} C(\boldsymbol{\varphi}_{ij}, \mathbf{x}_0) = \nabla_{\boldsymbol{\varphi}} S(\boldsymbol{\varphi}_{ij})$ we get

$$\mathbf{H}(\mathbf{x}_0) = \sum_{i,j} \frac{1}{D^2} \mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \nabla_{\boldsymbol{\varphi}} S(\boldsymbol{\varphi}_{ij}) \cdot [\mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \nabla_{\boldsymbol{\varphi}} S(\boldsymbol{\varphi}_{ij})]^T. \tag{8}$$

This matrix is positive semidefinite: If we write $\mathbf{q}_{ij} = (1/D)\mathbf{B}(\beta_i)^T \boldsymbol{\Gamma}(\gamma_j) \nabla_{\boldsymbol{\varphi}} S(\boldsymbol{\varphi}_{ij})$, we see that

$$\mathbf{a}^T \mathbf{H}(\mathbf{x}_0) \mathbf{a} = \mathbf{a}^T \left(\sum_{i,j} \mathbf{q}_{ij} \mathbf{q}_{ij}^T \right) \mathbf{a} = \sum_{i,j} (\mathbf{q}_{ij}^T \mathbf{a})^2 \geq 0$$

for arbitrary vectors $\mathbf{a}$. Again, to ensure strictly positive definiteness (hence invertibility), we have to test that $H_{xx}(\mathbf{x}_0) > 0$ and $\det \mathbf{H}(\mathbf{x}_0) > 0$. The Hessian will be strictly positive definite whenever there is a strict local minimum of the potential function at the snapshot position. In our simulations, these conditions were always fulfilled. We only determine $\mathbf{H}(\mathbf{x}_0)^{-1}$ once for each snapshot position and use it for all computations of the home vector according to (7). We also have to check for positive definiteness only once. It is interesting to note that the Hessian, when computed at the snapshot position, does not depend on second-order but only on first-order derivatives in images. This is an advantage since second-order derivatives are known to be susceptible to noise.

We determined $\mathbf{H}(\mathbf{x}_0)$ from (8) for a snapshot position in the center of the two virtual rooms (see Figs. 1 and 5, cutoff frequency 0.05). For the room with two textured walls on opposite sides, we get

$$\mathbf{H}(\mathbf{x}_0) = \begin{pmatrix} 0.079 & -0.026 \\ -0.026 & 0.250 \end{pmatrix}, \qquad \det \mathbf{H}(\mathbf{x}_0) = 0.019,$$

where the diagonal elements reveal that the image distance function estimated by the Newton-MFDID method is elongated along the x-direction (and slightly rotated since the off-diagonal elements are non-zero). For the room with four textured walls, we obtain

$$\mathbf{H}(\mathbf{x}_0) = \begin{pmatrix} 0.340 & 0.017 \\ 0.017 & 0.349 \end{pmatrix}, \qquad \det \mathbf{H}(\mathbf{x}_0) = 0.118,$$

where the diagonal elements are about equal which indicates that the image distance function is roughly isotropic (again, it is slightly rotated against the axes). In both cases, we see that $H_{xx}(\mathbf{x}_0) > 0$ and $\det \mathbf{H}(\mathbf{x}_0) > 0$, thus $\mathbf{H}(\mathbf{x}_0)$ is strictly positive definite.

All derivations above are expressed in angular coordinates $\boldsymbol{\varphi}_{ij}$. Versions of the gradient and the Hessian adapted for standard image processing operations in pixel coordinates can be derived as described in our previous paper (Möller and Vardy 2006).

7 DID compass

The DID and MFDID methods described in Sects. 3, 4, 5, and 6 all require the processed images to be aligned with respect to the same external coordinate system. Unfortunately, magnetic compasses which could be used for this purpose are notoriously unreliable in indoor environments. Zeil et al. (2003) demonstrated that image distances between two spatially separated images (one snapshot, one current view) exhibit a pronounced minimum under rotation (or in panoramic images, horizontal shift). Figure 4 shows an example of three difference functions obtained for one of our databases (see also Fig. 5) when the snapshot was rotated through −180° . . . 180°. The snapshot image was taken

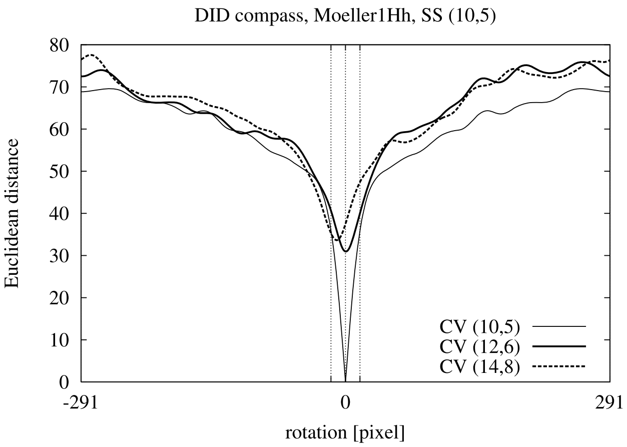


Fig. 4 DID compass: For a snapshot (SS) from grid position (10, 5) and three different current views (CV), the Euclidean distance under rotations of the snapshot image is shown. All images are taken from the database Moeller1Hh (image size 583 × 81). The two vertical dotted lines indicate the range of shifts used in our experiments (−16 . . . 16, corresponding to approximately −10° . . . 10°). Image pixels are in the range [0, 1]. Cutoff frequency 0.05

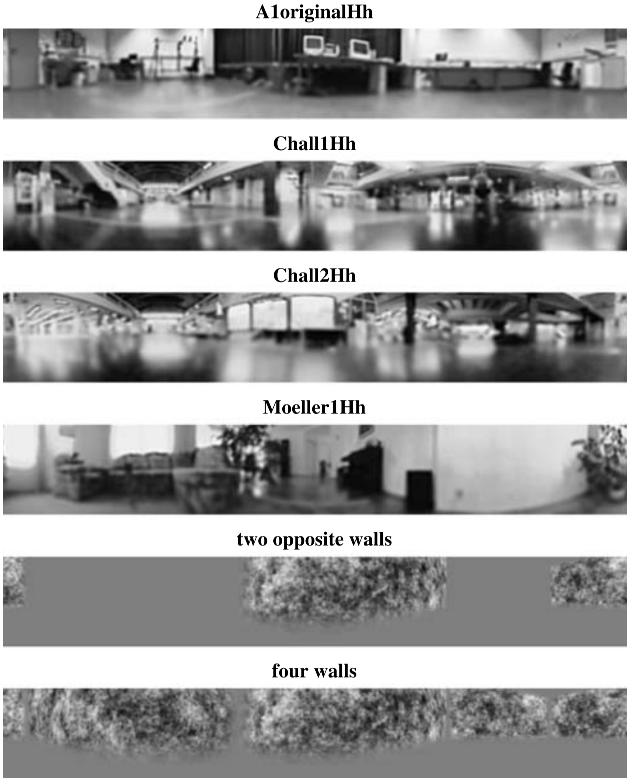


Fig. 5 Example images of the six databases, taken approximately in the center of the grid in the real-world databases, and at grid position $(3, 3)$ in the virtual databases. The images shown for the real-world databases are the versions with histogram equalization. Cutoff frequency 0.20

at the grid location $(10, 5)$, the current images at $(10, 5)$, $(12, 6)$, and $(14, 8)$. Since the database images are aligned, the minima of all curves are found close to zero rotation. When snapshot and current location coincide, the minimal distance drops to zero; when they are spatially separated, there is a local minimum with a distance above zero, but the corresponding shift is still close to the true relative orientation of the images.

8 Experiments

The main goal of our experiments is to demonstrate that the angular error between the home vector and the true home direction is smaller in the Newton-based versions of DID and MFDID than in the corresponding gradient-based methods; the focus lies on the MFDID methods since they are better suitable for practical applications. Our analysis is based on databases of real-world images which have been collected by a mobile robot with a panoramic camera (a catadioptric system which allows omnidirectional vision via a hyperbolic mirror). Experiments in image databases are an established technique which considerably simplifies the statistical analysis while allowing a direct transfer of the methods to true robot experiments. In addition, to highlight the influence of the environment on the method, we use two virtual databases where the images have been computed by rendering scenes with randomly textured walls. Each of the four real-world databases is used in two versions, with or without histogram equalization. Table 1 provides the grid size and grid spacing for all six databases; Fig. 5 shows example images. All images have the same size (561×81), except for the database Moeller1H (583×81); the horizon is located approximately at row 40.

Table 1 Description of the six databases used in our experiments

	Grid size	Spacing
A1originalH (lab room)	10 × 17	0.3 m
Chall1H (large hall)	10 × 20	0.5 m
Chall2H (large hall)	8 × 20	0.5 m
Moeller1H (living room)	22 × 11	0.1 m
Two opposite walls (virtual)	19 × 19	0.3 m
Four walls (virtual)	19 × 19	0.3 m

The real-world databases A1originalH, Chall1H, and Chall2H have been described in detail in our previous publications (Vardy and Möller 2005; Möller and Vardy 2006). These indoor databases have been collected under constant conditions of illumination (at night) with fixed gain and exposure time. Each line of the grid was collected while the robot was moving straight based only on odometry data. Small deviations in orientation and the resulting deviations from the desired grid positions were determined subsequently in a manual procedure by using distant landmarks as reference points. Using these calibration data, the images are brought into the same orientation, and estimates of the true positions are used for the analysis of angular errors, not just grid positions.

The indoor database Moeller1Hh was collected under varying conditions of illumination, but with a software mechanism that adjusted exposure time (and, under extremely dark conditions, also the gain) such that the average brightness of the image was kept constant. Moreover, the spatial resolution of this database is better (0.1 m) since the robot was visually guided by means of a second camera along a thread carrying distance markers (which is not visible in the database images). The positions and orientations were determined by analyzing the image of the thread. Also in this database, orientations are corrected in the preprocessing and true positions are used in the analysis.

The images of all four real-world databases are preprocessed in the following way: When histogram equalization is used, it is applied only to a ring-shaped region in the original camera image.[1] Afterwards the image is filtered

[1] We refer to databases with histogram equalization by appending a lower-case letter h to the database name.

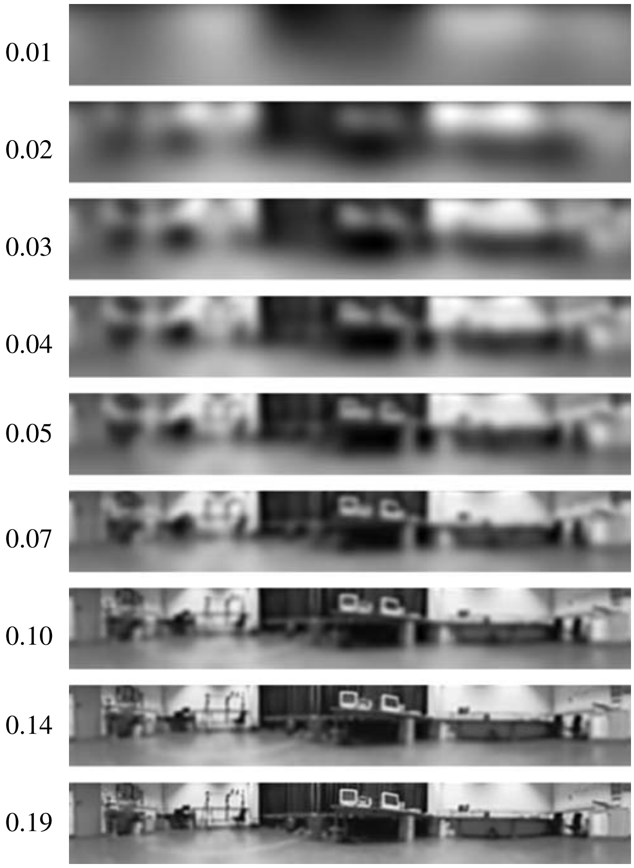


Fig. 6 Effect of varying cutoff frequency of the Butterworth filter on the preprocessed images, visualized for the database A1originalHh

with a Butterworth lowpass (3rd order) with varying relative cutoff frequency (0.01, 0.02, 0.03, 0.04, 0.05, 0.07, 0.10, 0.14, 0.19). From the lowpass-filtered image, a panoramic version (as shown in Fig. 5) is produced by a hyperbolic mapping. To give a visual impression of how different cutoff frequencies affect the final image, Fig. 6 shows nine preprocessed images taken from the same position in the database A1originalHh.

The virtual room used for the two remaining databases has a width and length of 6 m and a wall height of 3 m; the camera is looking at this scene from a height of 0.4 m. The random wall texture has an amplitude spectrum of $1/f$ (with f being the frequency) mimicking the spectral distribution in natural images (Ruderman and Bialek 1994; van der Schaaf and van Hateren 1996). Wall texture pixels have a size of 1 cm. High frequencies at the wall borders were avoided by a smooth transition between wall and gray background. To determine the intensity of a single pixel, the texture is averaged over a rectangular area of the wall determined by the pixel's angle of view (anti-aliasing). In the virtual databases, the Butterworth filter is applied to the panoramic images.

The angular error $\arccos(\mathbf{h}^T\hat{\mathbf{h}}) \in [0, \pi]$ between the true home direction $\mathbf{h}$ and the home vector $\hat{\mathbf{h}}$ (which is either $\mathbf{h}_g$, $\mathbf{h}_n$, $\mathbf{h}_g^*$ or $\mathbf{h}_n^*$) is determined for each combination of a snapshot position and a second position in the neighborhood, and the median of the angular errors is used for the comparison between DID and Newton-DID and between MFDID and Newton-MFDID. All grid positions are used as snapshot positions. Since Newton's method is only applicable in the vicinity of the goal position, we compute home vectors only for the 20 grid positions (up to a distance of 2.5 grid units from the snapshot) shown in Fig. 2 (less at the borders of the database grids). This gives between approximately 2 800 and 6 400 image pairs depending on the size of the database. We decided to use the same local neighborhood in the grid despite the differences in grid spacing, since the denser grids were collected in environments where the landmark features are also found at smaller distances from the robot.

For the statistical tests, we use bootstrapping with BC_α correction (Efron and Tibshirani 1998). Bootstrapping has the advantage that it does not rest on assumptions about how the data are distributed. For a given image database, we compare the median angular error for the two methods we wish to compare: the best DID method (over the parameters histogram equalization on/off and cutoff frequency) vs. the best Newton-DID method, and the best MFDID method vs. the best Newton-MFDID method. The data set comprises the angular errors of the two methods for each pair of image locations. This data set is re-sampled 10 000 times, the median angular errors are computed, and the confidence interval is determined from the resulting distribution of the difference of the median angular errors. We test whether zero lies outside the confidence interval obtained for $\alpha = 1\%$ (two-sided test) from the percentiles of this distribution, in which case the median angular errors are significantly different.

9 Results

9.1 DID vs. Newton-DID

First we analyze whether Newton-DID improves the quality of home vectors over DID (see Sects. 3 and 4). We restrict this experiment to the database Moeller1H, since the performance of the DID method degrades with increasing spatial distance between the sampling points (see Möller and Vardy 2006, Fig. 11); the database Moeller1H has the smallest grid spacing (0.1 m) of our databases (see Table 1). Figure 7 shows the results. It is apparent that the performance of the two methods varies with the cutoff frequency of the Butterworth filter: The median angular error decreases with increasing cutoff frequency and then stays approximately

Table 2 Results of the statistical tests applied to the best gradient-based and the best Newton-based DID or MFDID method for each database. The parameters histogram equalization (hist-eq) and Butterworth cutoff frequency (cutoff), and the median angular error (med. AE) are given for each method. The last column shows whether the median angular errors of the two compared methods differ significantly (** = significant with $\alpha = 1\%$, n.s. = not significant)

		Gradient			Newton			Significant
Method	Database	hist-eq	cutoff	med. AE	hist-eq	cutoff	med. AE	$\alpha = 1\%$
DID	Moeller1H	yes	0.19	0.179	yes	0.07	0.106	**
MFDID	A1originalH	yes	0.02	0.130	no	0.03	0.082	**
MFDID	Chall1H	yes	0.14	0.103	no	0.05	0.075	**
MFDID	Chall2H	yes	0.14	0.093	no	0.10	0.090	n.s.
MFDID	Moeller1H	yes	0.05	0.147	yes	0.03	0.093	**
MFDID	two opposite walls	no	0.01	0.339	no	0.05	0.172	**
MFDID	four walls	no	0.14	0.132	no	0.05	0.114	**

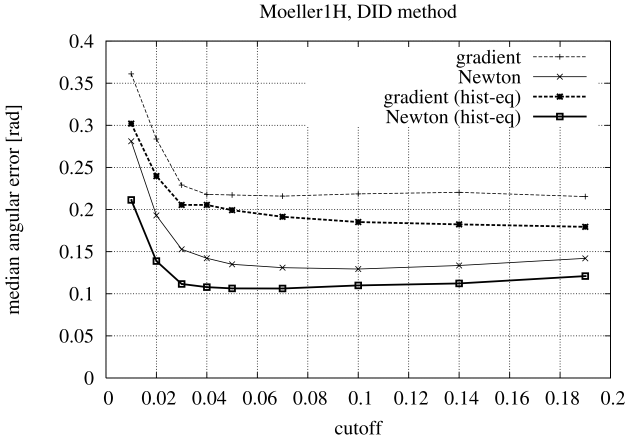


Fig. 7 DID vs. Newton-DID: Median angular errors for the database Moeller1H, depending on the cutoff frequency. Thin lines: without histogram equalization, thick lines: with histogram equalization. Dashed lines: gradient method (DID), solid lines: Newton method (Newton-DID)

constant. Histogram equalization seems to improve the performance for both methods; we will discuss this in Sect. 10. Table 2 compares the best result over the parameters cutoff frequency and histogram equalization (on/off) for DID and Newton-DID (first line). The statistical test shows that Newton-DID performs significantly better than DID.

9.2 MFDID vs. Newton-MFDID

Now we closely examine the performance of the matched-filter methods MFDID and Newton-MFDID. Figure 2 shows an example of the home vectors obtained from MFDID (gray, short) and Newton-MFDID (long, black) for a position in the center of the virtual room with two walls. It is clearly visible that, for this position, Newton-MFDID produces home vectors with smaller angular deviation from the true home direction. The median angular errors over all pairs of image locations are visualized in Fig. 8.

In all databases, the cutoff frequency of the Butterworth lowpass influences the performance of both MFDID and Newton-MFDID. In some databases (A1originalH, A1originalHh, Chall1H; see footnote 1), the median angular error exhibits a distinctive optimum at low cutoff frequencies (strongly lowpass-filtered images), in other databases, the performance improves drastically with rising cutoff frequency up to a certain point, but stays approximately constant for higher cutoff frequencies. The effect of histogram equalization is mixed for the different databases. It improves the performance of the gradient method in A1originalHh, Chall1Hh, and Chall2Hh, and reduces the performance of the Newton method in Chall1Hh, but has a relatively small effect on Moeller1Hh.

Table 2 lists the performance of the best gradient method and the best Newton method for each database. In three of the real-world databases, Newton-MFDID performs significantly better than the gradient-based MFDID method. In the remaining real-world database (Chall2H), the two methods do not differ significantly. The clearest gain of Newton-MFDID is noticeable for the virtual database with two opposite walls, but also in the four-walled virtual room Newton-MFDID fares significantly better than MFDID (which is probably due to anisotropies introduced by the random wall texture).

In addition, we tested how an increase in the distance between snapshot and current view affects the performance of gradient and Newton method. For the database A1originalHh (with histogram equalization), the median angular error was determined for a neighborhood of 96 points around each snapshot (all neighbors up to a radius of 5.5 grid units).[2] Table 3 compares the best results for 20 neigh-

[2] Note that a radius of 5.5 grid units constitutes a large area covering the whole width of the grid in the database A1originalHh.

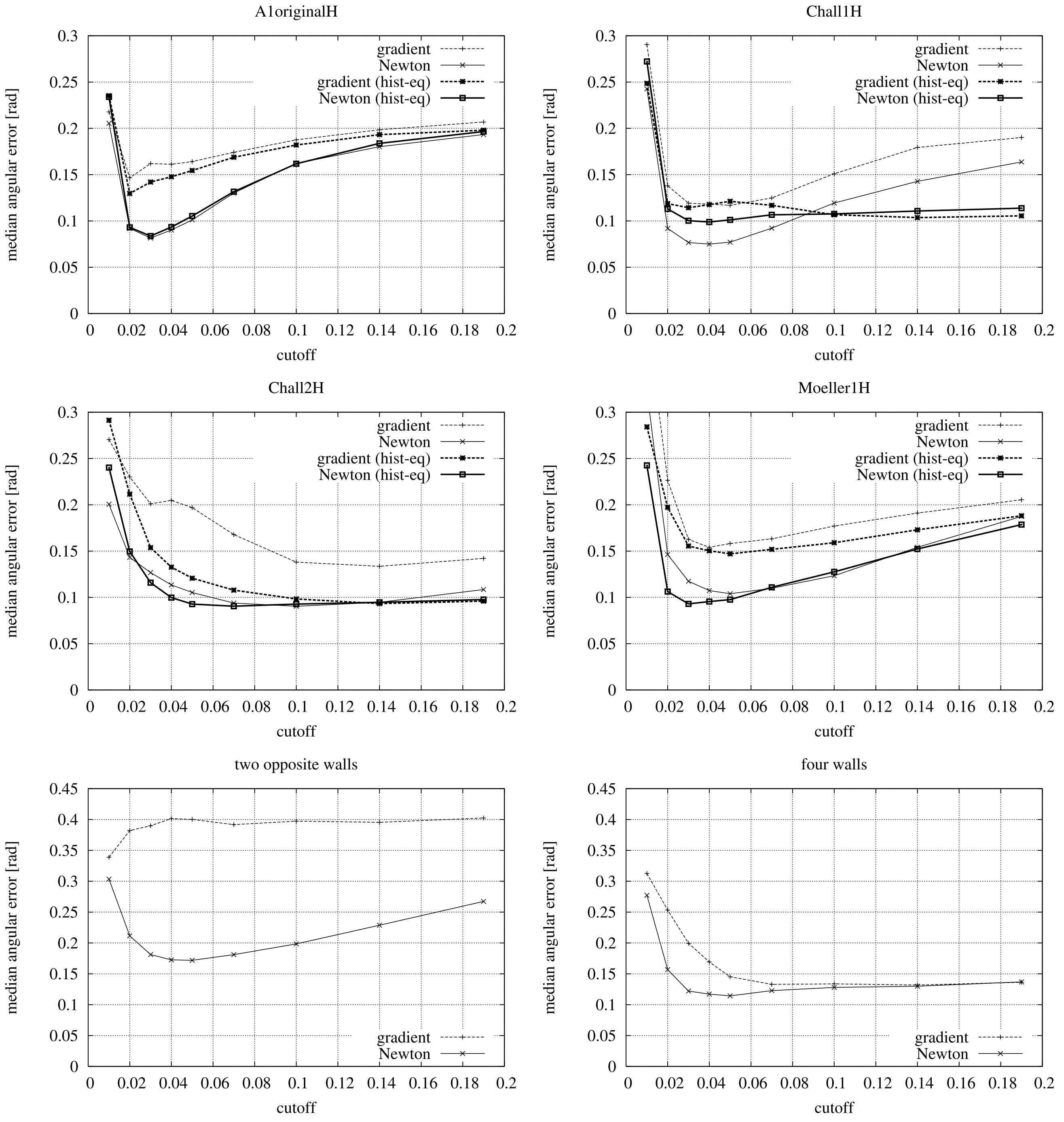


Fig. 8 MFDID vs. Newton-MFDID: Median angular errors for all six databases, depending on the cutoff frequency. *Thin lines*: without histogram equalization, *thick lines*: with histogram equalization. *Dashed lines*: gradient method (MFDID), *solid lines*: Newton method (Newton-MFDID)

boring positions (2.5 grid units, see Figs. 2 and 8) and 96 neighboring positions (5.5 grid units). The performance of both gradient and Newton method is reduced for the larger neighborhood, but Newton-MFDID still exhibits a significantly smaller median angular error, although the gain relative to the gradient method is diminished.

9.3 Influence of compass errors

All DID methods rely on the assumption that the images are aligned with respect to the same external frame of reference. While all images in the database are available in the same orientation, the orientation needs to be estimated in a true robotic application by some type of compass which will be affected by errors. Here we analyze the extent to which the matched-filter methods tolerate errors introduced by the DID compass (see Sect. 7). We restrict this analysis to the real-world databases with histogram equalization.

In a first series of experiments, we shifted the snapshot through −16...16 pixels (corresponding to approximately −10°...10°), and analyzed the effect of this compass error onto the median angular error for different cutoff frequencies (again, the current views are taken from the vicinity of the snapshot location). Figure 10 (left) shows the results for the database A1originalHh and the Newton-MFDID method; the results for all other databases and for the gradient-based MFDID were qualitatively the same. The most important insight is that increasing cutoff frequencies reduce the tolerance against compass errors: The median angular error rises faster with increasing compass error for larger cutoff frequencies. Thus, for a practical application, not only the performance for an ideal compass (zero shift) should be considered when the cutoff frequency is chosen, but also the tolerance against compass errors.

In a second series of experiments, we determined the average compass error (absolute value) of the DID compass for the four databases and for varying cutoff frequencies. Figure 9 visualizes the histogram of the optimal shifts found by the DID compass (the correct shift is 0). Although the search range (−16...16) used here appears to be relatively small (see also Fig. 4), the majority of optimal shifts lies within this range, and we notice only a small fraction of outliers pooled at the borders of the histogram (bins −16 and 16). The results for all databases and cutoff frequencies are shown in Fig. 10 (right). It is apparent that with the best parameters, the average error is below 3 pixels, usually even smaller (except for the database A1originalHh). When we compare this range of errors to the tolerance of the MF-DID methods (see arrows in Fig. 10), we get the visual impression that the home vector accuracy will only be affected mildly by these errors.

We verify this impression by a third series of experiments where the images are aligned by means of the DID compass before the MFDID methods are applied to determine the home vector. We use a cutoff frequency of 0.04 for the DID compass which seems to be a good compromise over all databases (see black triangle in Fig. 10,

Table 3 Comparison of gradient and Newton method for the database A1originalHh and different maximal distance (2.5, 5.5 grid units) between snapshot and current views. Abbreviations are explained in the caption of Table 2

	Gradient		Newton		Signif.
dist.	cutoff	med. AE	cutoff	med. AE	(1%)
2.5	0.02	0.130	0.03	0.084	**
5.5	0.02	0.151	0.02	0.115	**

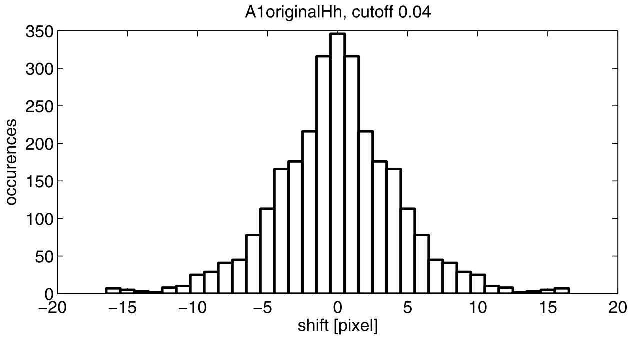


Fig. 9 Histogram of optimal shift values obtained from the DID compass for the database A1originalHh with cutoff 0.04

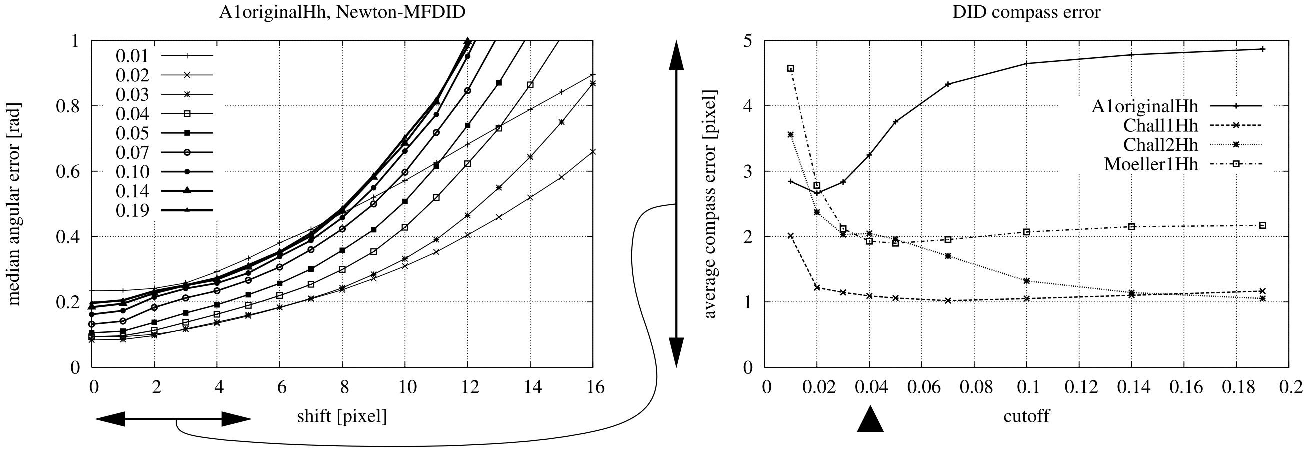


Fig. 10 *Left*: Median angular error of Newton-MFDID in the database A1originalHh for varying cutoff frequencies under compass errors (i.e. horizontal shift of the panoramic snapshot image). *Right:* Average compass error for four databases and varying cutoff frequencies. The *black triangle* indicates the cutoff frequency used for the analysis in Fig. 11

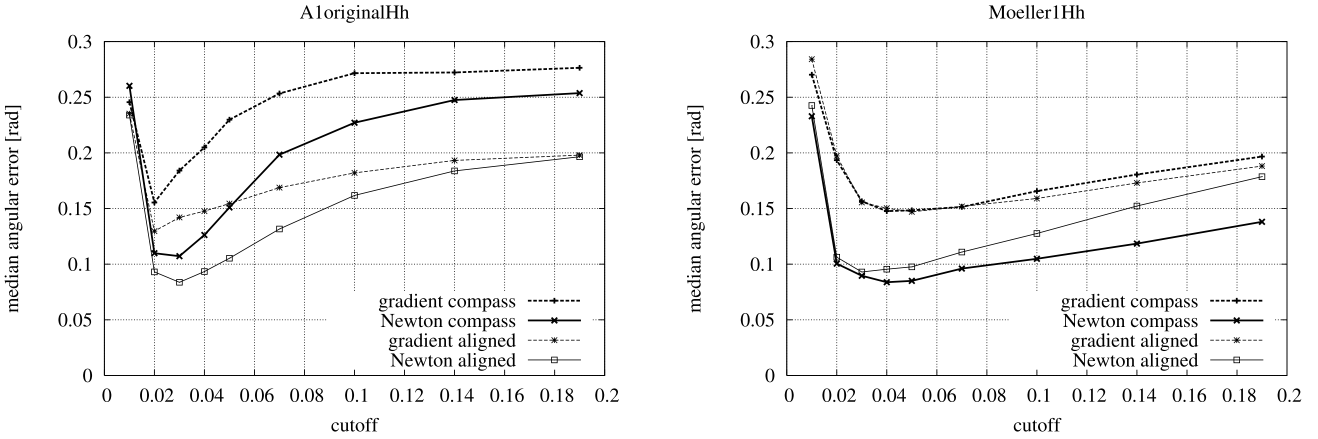


Fig. 11 Performance of MFDID and Newton-MFDID when the images are aligned by the DID compass. *Left*: Database A1originalHh. *Right:* Database Moeller1Hh. *Thin lines*: images are aligned (constant orientation), *thick lines*: image orientation is determined from the DID compass. *Dashed lines*: gradient-based MFDID, *solid lines*: Newton-MFDID

right). Figure 11 shows that for low cutoff frequencies, the performance of the methods is only mildly degrading in the database A1originalHh when the DID compass is used (compared to images which are presented in the same orientation). For larger cutoff frequencies, the performance is more strongly affected (as would be expected from the compass error for this database shown in Fig. 10, right). In the database Moeller1Hh, the performance of MFDID is hardly affected, and Newton-MFDID works even better with the DID compass than with perfect alignment which is somewhat surprising.

10 Discussion

10.1 Summary of results

The results allow the overall statement that Newton-MFDID improves the performance over MFDID in all but one of our databases and has no negative effect on the other database. We notice that in some cases (Chall2H) the advantage of Newton-MFDID is lost when histogram equalization is applied to the images. Our explanation is that since histogram equalization accentuates hidden features in some regions of the image, it reduces the anisotropy of the landmark distribution and thus improves the performance of the gradient method. Why it sometimes also degrades the performance of Newton-MFDID (as in Chall1H) is not clear.

The virtual databases clearly demonstrate the expected effect: If the landmark distribution is anisotropic (two opposite walls), the gradient home vectors exhibit large deviations from the true home direction, and the gain by Newton-MFDID is substantial. For almost isotropic scenes (four walls), the gradient vectors are of good quality and the improvement achieved by Newton-MFDID is small (although statistically significant). Also for three of the real-world databases the gain achieved by the Newton-MFDID method is significant. Why Newton-MFDID improves the performance is not always apparent from the visual scene. In the database Moeller1H, there is a large white, featureless wall on one side of the room and more features on the other side (see Fig. 5), but such an anisotropy is not immediately visible in the databases A1originalH and Chall1H. Newton-MFDID maintains a lower error than MFDID even at large distances from the snapshot position (see Table 3).

What Newton-MFDID cannot mend, of course, are anisotropies introduced by large deviations in the distances to the landmarks, since both MFDID and Newton-MFDID rest on an equal-distance assumption. It should also be pointed out that anisotropies affect all DID-based methods, but not correspondence methods: When correspondence vectors are transformed into movement vectors (“vector mapping”, see Vardy and Möller 2005), all movement vectors point towards the goal location; even without distance information, the direction of the individual movement vectors and thus also their average is correct.

10.2 Applications

In the vicinity of the target location, the Newton-MFDID method produces more accurate home vectors than the MFDID method. This improved performance can be decisive for several applications in the context of topological navigation approaches. The first application concerns the process of homing based on the graph-like representation of the environment in a topological map. After a shortest path has been planned from the starting node to the goal node, the robot has to follow the path from node to node: When it has reached a node, it selects the next node on the path as target node and approaches this node based on a local visual homing method. Since nodes linked by an edge are usually supposed to be in close vicinity (Franz et al. 1998), the Newton-MFDID method is well suitable for this purpose. It would allow the robot to approach the next target node on a straight rather than a curved path, thus reducing the overall distance traveled.

Second, improved home vector accuracy may also be a prerequisite for methods which add metrical information to topological maps. In the pure form of topological maps, nodes are only annotated with sensory information that was collected when the robot visited the corresponding location, but not with information on the metrical position of the location. However, for some applications it is beneficial if the metrical position can subsequently be assigned to the graph nodes, for example if the map has to be visualized for a human operator, or if the robot has to determine unexplored regions. Several metrization methods have been suggested which exploit information like metrical distance or angular relations between nodes (Hafner 2000; Hübner and Mallot 2002; Hübner 2005). In the context of visual navigation, local home vectors could provide the direction to neighboring nodes, or could be used to determine the distance by triangulation procedures. Clearly, the accuracy of the home vectors is crucial for such methods.

Finally we would like to mention another potential application of the Newton-MFDID method in the context of navigation in topological maps. When an agent is attempting to reach a goal location through a series of intermediate locations with stored snapshots, it has to use some criterion to decide whether a location has been reached, before it can switch to the next snapshot in the sequence. In the DID framework, the criterion could be that a distance measure like the SSE drops below a threshold, but due to the distorted iso-potential lines as visible in Fig. 1 (top), the threshold would be reached in different distance from the location when the agent approaches it from different directions. This distortion can be compensated for when the length of the Newton-MFDID home vector is used instead of the potential; ideally, this measure will be proportional to the distance from the location. The effect is clearly visible in Fig. 12 which compares the iso-lines of the image distance (Euclidean) with those of the length of the Newton-MFDID home vector for the virtual database with two opposite walls.

However, due to the unknown average distance to the landmarks, the vector length will not encode the absolute distance to the location, thus there will still be the difficulty to define an *absolute* termination threshold. For this purpose, coarse estimates of the average landmark distance might be inserted as D into the equations. Figure 13 visualizes the length of the Newton-MFDID home vector for all 19 × 19 grid points when an average distance D of 3.6 m is assumed (the perpendicular distance to the walls is 3 m); it is apparent that the vector length provides a reasonably good estimate of the true distance when the goal is nearby. The average distance could be estimated by analyzing the relation between true distance and home vector length on short-distance exploratory excursions where odometry data are sufficiently precise to allow estimation of the true distance to the goal.

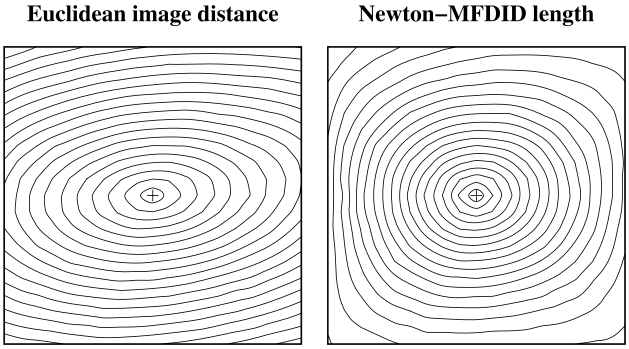


Fig. 12 Euclidean image distance (*left*) and length of the Newton-MFDID home vector $\mathbf{h}_n^*(\mathbf{x})$ from (7) (*right*) for a region in the center of the virtual room with two opposite walls. The snapshot image was taken at the position marked with the cross in the center of the room, and the current views come from a grid of 19 × 19 images within the square; the square has a side length of 90 cm. Cutoff frequency: 0.05

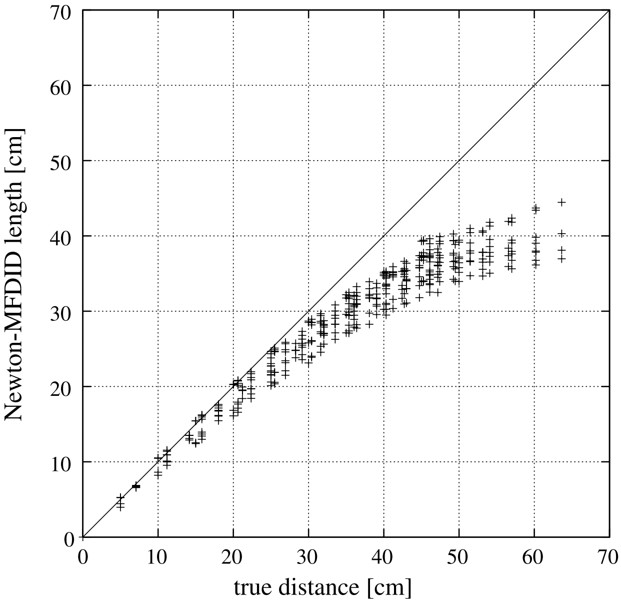


Fig. 13 Length of the Newton-MFDID home vector vs. true distance to the goal location for an estimated average landmark distance of 3.6 m (see Fig. 12, right). Cutoff frequency: 0.05

11 Conclusion

In environments where the distribution of landmark features is anisotropic, Newton's method can improve the performance of MFDID in the vicinity of the snapshot location; in no case did the Newton method significantly reduce the performance compared to the original, gradient-based MFDID method. Thus, Newton-MFDID should be used instead of MFDID both for the homing process itself and for other applications such as triangulation and arrival detection.

Acknowledgements We are grateful to Alexej Schatz who provided his Matlab implementation of the bootstrapping method, and to Wolfram Schenck for his advice on the statistical tests. Frank Röben and Lorenz Gerstmayr contributed to the collection of one of the databases. This work was supported by the Deutsche Forschungsgemeinschaft (grant no. MO 1037/2-1).

Appendix 1 Modified Newton method

In the modified Newton method used in this paper, the Taylor approximation is done at the goal location $\mathbf{x}_0$ (where we have $\nabla_{\mathbf{x}} p(\mathbf{x}_0) = \mathbf{0}$):

$$\begin{aligned} p(\mathbf{x}) &\approx p(\mathbf{x}_0) + \nabla_{\mathbf{x}}^T p(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) \\ &\quad + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) \\ &= p(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0). \end{aligned}$$

The gradient of this expression is

$$\nabla_{\mathbf{x}} p(\mathbf{x}) = \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0).$$

We introduce the home vector $\mathbf{h} = \mathbf{x}_0 - \mathbf{x}$ and obtain

$$\mathbf{h} = -\mathbf{H}(\mathbf{x}_0)^{-1} \nabla_{\mathbf{x}} p(\mathbf{x}).$$

In this version of Newton's method, the gradient is determined at the current position $\mathbf{x}$, but the Hessian is computed at the goal position $\mathbf{x}_0$.

References

Binding, D., & Labrosse, F. (2006). Visual local navigation using warped panoramic images. In *Proceedings of towards autonomous robotic systems*. University of Surrey, Guildford, UK.

Chong, E. K. P., & Żak, S. H. (2001). *An introduction to optimization* (2nd ed.). New York: Wiley.

Efron, B., & Tibshirani, R. J. (1998). *An introduction to the bootstrap*. New York: Chapman & Hall/CRC.

Franz, M. O., & Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems, special issue: Biomimetic Robots*, *30*(1–2), 133–153.

Franz, M. O., Schölkopf, B., Mallot, H. A., & Bülthoff, H. H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, *5*(1), 111–125.

Hafner, V. V. (2000). Cognitive maps for navigation in open environments. In *Proceedings 6th international conference on intelligent autonomous systems (IAS-6)* (pp. 801–808), Venice. Amsterdam: IOS.

Hübner, W. (2005). *From homing behavior to cognitive mapping: integration of egocentric pose relations and allocentric landmark information in a graph model*. PhD thesis, Fachbereich 3 (Mathematik & Informatik), Universität Bremen.

Hübner, W., & Mallot, H. A. (2002). Integration of metric place relations in a landmark graph. In J. R. Dorronsoro (Ed.), *Lecture notes in computer science: Vol. 2415. Proceedings of the international conference on artificial neural networks (ICANN)* (pp. 825–830). Berlin: Springer.

Möller, R., & Vardy, A. (2006). Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, *95*(5), 413–430.

Ruderman, D. L., & Bialek, W. (1994). Statistics of natural images: scaling in the woods. *Physical Review Letters*, *73*(6), 814–817.

van der Schaaf, A., & van Hateren, J. H. (1996). Modelling the power spectra of natural images: statistics and information. *Vision Research*, *36*(17), 2759–2770.

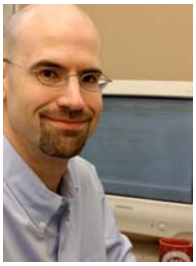
Vardy, A. (2006). Long-range visual homing. In *Proceedings of IEEE international conference on robotics and biomimetics, ROBIO'06* (pp. 220–226).

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

Zampoglou, M., Szenher, M., & Webb, B. (2006). Adaptation of controllers for image-based homing. *Adaptive Behavior*, *14*(4), 381–399.

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

Ralf Möller heads the Computer Engineering Group at the Faculty of Technology of Bielefeld University. His research interests include behavior-based approaches to visual cognition, visual navigation, biorobotics, neuromorphic systems, and parallel computation.



Andrew Vardy is an Assistant Professor jointly appointed to the Department of Computer Science and the Faculty of Engineering and Applied Science at Memorial University of Newfoundland, St John's, Canada. His primary research topic is visual robot navigation, especially approaches inspired by the visual navigation abilities of animals.

Sven Kreft studied computer science at University of Bielefeld. In 2007 he received a M.Sc. degree in computer science from the Faculty of Technology of Bielefeld University. Since 2007 he is Member of the Virtual Reality group at the Heinz Nixdorf Institute and works as a Ph.D. candidate in the Department of Computer Integrated Manufacturing at the University of Paderborn. His research interest is in 3D computer graphics and in the development of Virtual Reality applications, specially in the field of Virtual Prototyping.

Sebastian Ruwisch was a student at the Faculty of Technology of University Bielefeld and worked as a student assistant in the Computer Engineering Group. He finished his diploma thesis in Applied Computer Science in the Natural Sciences in march 2007.