An Orientation Invariant Visual Homing Algorithm

David Churchill[a], Andrew Vardy[b,*]

[a] *Department of Computer Science, University of Alberta, Edmonton, Canada*
[b] *Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada*

Abstract

Visual homing is the ability of an agent to return to a goal position by comparing the currently viewed image with an image captured at the goal, known as the snapshot image. In this paper we present additional mathematical justification and experimental results for the visual homing algorithm first presented in [1]. This algorithm, known as Homing in Scale Space, is far less constrained than existing methods in that it can infer the direction of translation without any estimation of the direction of rotation. Thus, it does not require the current and snapshot images to be captured from the same orientation (a limitation of some existing methods). The algorithm is novel in its use of the scale change of SIFT features as an indication of the change in the feature's distance from the robot. We present results on a variety of image databases and on live robot trials.

1. Introduction

Visual homing (VH) provides the ability for an agent to return to a previously visited position by comparing the currently viewed image with a remembered image captured at the reference position. This allows the agent to return to the reference position from any nearby point with a sufficient degree of visual similarity. In this paper we provide formal justification for the Homing in Scale Space algorithm first proposed in [1]. We also present additional experimental results that demonstrate the algorithm's effectiveness for a variety of different environments and on live robot trials.

VH has been studied both as a model for local animal navigation and as a tool for local robot navigation. A particular model of the homing behaviour of honeybees known as the snapshot model was proposed by Cartwright and Collett [2, 3]. This model proposes that honeybees can return to important locations in their environment by pairing visual features between the current image and an image stored at the goal position known as the snapshot image. Disparities in both the position and size of features in the current image from the snapshot image are used to compute correcting vectors. These vectors are then summed to produce an overall home vector. This general strategy pervades much of the work on VH from

*Corresponding author
Email address: `dave.churchill@gmail.com` (David Churchill)
URL: `http://www.cs.mun.ca/~av` (Andrew Vardy)

both the biological and robotics communities, including the work described in this paper. However, as far as we are aware the algorithm presented here is the first since Cartwright and Collett's snapshot model to make explicit use of the disparity in apparent size of visual features. The biological community have proposed variants of the snapshot model, as well as alternative homing strategies for a variety of species including honeybees [4, 2, 3], ants [5, 6], rats [7] and humans [8]. For social insects such as bees and ants it has been argued that visual homing (sometimes referred to as 'image matching') is a crucial component in their overall navigational strategy [9].

In robotics, a visual homing algorithm serves the purpose of a 'local control strategy' which Kuipers and Byun described as "how a robot can follow the link connecting two distinctive places" [10]. The chief limitation is that it can only be applied in the immediate neighbourhood of the goal location. There must be sufficient similarity between the current image and the goal image for an accurate home vector to be computed. If the goal locations are spaced closely together and in sequence then VH can be used as a means of executing learned routes through an environment [11, 12, 13]. If the goal locations are distributed throughout the environment, they can be treated as nodes in a graph. This representation is known as a topological map. To navigate using such a map requires a localization system that combines sensory information with a model of the robot's motion. VH then fills the role of moving the robot between connected nodes. It can also be used in the discovery of new edges between nodes [14]. This approach falls under the category of topological simultaneous localization and mapping (SLAM). VH has been employed by a variety of researchers on topological SLAM [14, 15, 16, 17].

Visual homing can be considered a form of qualitative navigation, in the sense of Dai and Lawton where spatial learning and path planning proceed "in the absence of a single global coordinate system" [18]. This is in contrast with most work on grid-based or metric SLAM where the production of a single coordinate frame map is the ultimate goal. The difference lies in the degree of accuracy required to achieve the task at hand. It is possible to visually home to a previously visited position even with inaccurate information about its direction. As long as the difference between the robot's direction of movement and the ideal direction is less than 90^o the robot will eventually reach home [19] (although naturally we strive for higher accuracy). In the SLAM framework reaching a desired pose requires an accurate map, accurate localization of the robot within the map, and a further path planning stage. Methods of qualitative navigation such as visual homing are pursued because they offer the possibility of robust navigation with low computational cost.

The next section considers related work on the visual homing problem. We then present the mathematical formulation for the Homing in Scale Space algorithm. This is followed by a discussion of our experimental methods and results. We conclude with a discussion of these results and suggestions for future work.

2. Related Work

Existing methods for visual based homing can be classified as either holistic or correspondence-based [20]. In the next two sections we will discuss these two classes of homing algorithms.

2.1. Holistic Methods

Holistic methods rely on comparisons between images as a whole. An example of a holistic method is the method of Zeil et al. who posit a simple distance metric between images and implement homing as gradient descent in the space of this distance metric [21]. This method, while elegant in its simplicity, relies on the existence of a monotonic relationship between image distance and spatial distance. It also requires small exploratory movements of the robot in order to determine the gradient of the image distance function. Möller and Vardy described an alternative method based on gradient descent that removes the need for exploratory movements prior to computing a home vector [20].

Another holistic method is the so-called *warping method* of Franz et al. [19]. We present this method in some detail as it used as a benchmark for comparison with our method. The warping method searches for the parameters of motion which make the warped snapshot image most similar to the current image. A warped snapshot image is generated by transforming the snapshot image *as if* the robot had actually moved according to the given motion parameters. To make this transformation possible the assumption is made that all objects are equidistant from the goal. This assumption is rarely satisfied in practise. However, in environments where the objects are all relatively distant from the goal it provides a reasonable method of predicting the image that would result from small movements of the robot. A precise prediction would require *a priori* information on the structure of the environment, which is presumed not to be available in this context. The robot's movement is described by three parameters: α is the direction the robot has moved away from the goal, ψ is the change in orientation, and ν characterizes the distance to the goal relative to an assumed average landmark distance (see [19] for details). The snapshot image is warped by iterating over a discretized set of possible values for the movement parameters (α,ψ,ν). This search is tractable because it operates on one-dimensional images, which are sampled from the centre rows of two-dimensional images captured from the omnidirectional camera system. Despite the clearly unrealistic nature of the assumption that all landmarks are of equal distance from the snapshot, the warping method has been found to perform robustly in various indoor environments and has emerged as a standard for comparison for various visual homing methods [22, 23]. For this reason we utilize the warping method to benchmark the performance of our algorithm.

There has been notable recent progress by Möller in extending the warping algorithm to operate directly on two-dimensional images [24] and in relaxing the assumption that all landmarks lie at an equal distance from the snapshot location [25]. Comparison of our method with these newer variants of the warping framework is planned for future work.

2.2. Correspondence Methods

Correspondence based homing methods utilize feature detection and matching algorithms to form a set of correspondence vectors between the snapshot and current images. These vectors give the shift of the features in image space, known as the image flow field (c.f. 1). The flow field formed by these correspondence vectors is then interpreted to yield the direction of motion. These flow fields comprise both robot translation as well as rotation. The separation of these two components of motion can be difficult, therefore most correspondence methods posit the additional assumption that all images are counter-rotated to the same compass orientation prior to calculating homing direction. This process requires some form of compass, or a search for the change in orientation which would minimize the difference between the two images [21, 26, 27].

Vardy and Möller investigated the use of both matching and differential methods of optic flow for visual homing [22]. They determined that if both snapshot and current images were captured from the same orientation that the direction of translation could be computed analytically from a single correspondence. The optic flow techniques they used could produce dense flow fields, with home direction estimates produced for each vector in the flow field. The resulting home vector estimates were summed, which induced a cancellation of errors and resulted in very accurate and robust visual homing.

Various type of features have been utilized for determining correspondences, ranging in sophistication from raw image windows [22] to descriptors based on the Fourier-Mellin transform [28]. Other feature types which have been used include Harris corners [29], distinctive landmarks [11], and high contrast features [2, 30, 31]. Recently, Scale Invariant Feature Transform (SIFT) features have gained great popularity in many areas of computer vision and robotics due to the stability of their descriptor vectors with respect to changes in scaling, rotation, and illumination [32]. SIFT features have also been used to perform localization and visual homing [33, 34, 35, 16, 36].

Pons et al. [35] use SIFT features in order to recover image orientation before implementing the strategy of Vardy and Möller [22]. They search for the mode of the horizontal component of correspondence vectors as an indicator of the rotational component of motion. This technique is similar to one proposed by Röfer which sorts the horizontal shifts of all features and determines the value that would make the sign of half of the shifts positive and the other half negative [37].

Briggs et al. [34] deviate from the standard two-dimensional application of SIFT feature detection by utilizing one-dimensional images in order to reduce processing time and memory. Using the snapshot and current view images as the axes of a graph, images are matched using SIFT features and the resulting correspondence curve is plotted. The direction of motion required to return to the goal is then extracted from this matching curve. This technique has much in common with that of the original warping method [19] and its more recent two-dimensional variants [24, 25].

The method we present is similar to the correspondence methods described above in that it relies upon finding correspondences between features. However, our interpretation of the resulting correspondences is markedly different. Consider the flow field for pure translation of an agent equipped with an omnidirectional camera. The field has a characteristic structure with foci of expansion and contraction separated by 180° (see Figure 1). If objects are distributed uniformly in the environment, roughly half of them will appear to have expanded, while the remaining half will appear to contract. Typical correspondence methods consider how the features have shifted but not whether they have expanded or contracted. The problem is that in the presence of rotation it becomes much more difficult to determine the home direction from feature shifts. Hence, the two-stage process referred to above. However,

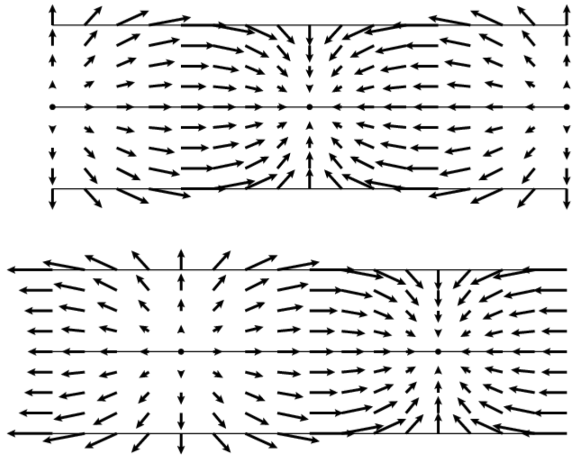


Figure 1: Ideal flow field for pure translation in a panoramic image [20].

whether a feature has changed in scale is independent of any change in orientation between the two views. We utilize the change in scale of corresponding SIFT features to move towards contracted features and away from expanded features.

3. Homing in Scale Space

3.1. Notation

The robot's current position and the snapshot (i.e. goal) position will be represented as position vectors $\boldsymbol{c}$ and $\boldsymbol{s}$ respectively. Let C and S represent the images captured from these positions. Features extracted from an image will be denoted with the same symbol, with a superscript giving the index of the feature. For example, S^j indicates the j^{th} feature extracted from the snapshot image.

One requirement of our method is that the direction of translation be visible within the robot's field of view. Therefore we utilize panoramic images that provide an omnidirectional field of view in the horizontal direction (c.f. 3(a)).

We will refer to our method as Homing in Scale Space or HiSS.

3.2. Visual Homing

If $\boldsymbol{c}$ and $\boldsymbol{s}$ lie within the same plane, then the ideal movement from $\boldsymbol{c}$ to $\boldsymbol{s}$ can be described by the home direction α and distance r (see figure 2). Some visual homing algorithms (e.g. [22, 35]) require the change in robot orientation ψ to be known prior to computing either α or r. The algorithm presented here has no such requirement. The method for estimating α is presented below. In the experimental section we consider a variety of techniques for estimating r.

If both α and r are known then the robot can move to its goal in a single step. However, due to the unknown scale of the environment it is often more difficult to obtain an estimate of r than of α. If only α is known then homing can still be achieved by making small steps in the direction of α. This requires some sort of similarity measure to determine when the robot has arrived at $\boldsymbol{s}$. In our experiments on live robot homing, we consistently underestimate r so that the robot moves towards the goal in smaller and smaller steps—a technique that prevents excessive oscillation around the goal.

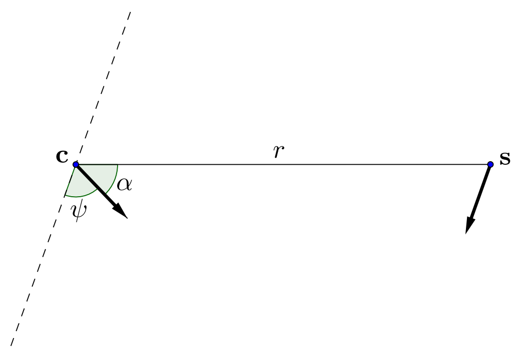


Figure 2: The unknown quantities in the visual homing problem. Thick arrows indicate the forwards orientation of the robot at $\boldsymbol{c}$ and $\boldsymbol{s}$. The dotted line through $\boldsymbol{c}$ is parallel to the robot's orientation at $\boldsymbol{s}$.

3.3. Feature Scale Change

In the description of our method below, we make geometric arguments on the basis of whether a perceived feature has expanded or contracted. That is, whether the object that generated the feature is closer or further from the robot at the current position than at some reference position. As opposed to estimating the distance to the feature, we use the change in the scale parameter of SIFT features to indicate whether the feature has expanded or contracted. Consider C^j the j^{th} feature extracted from the current image:

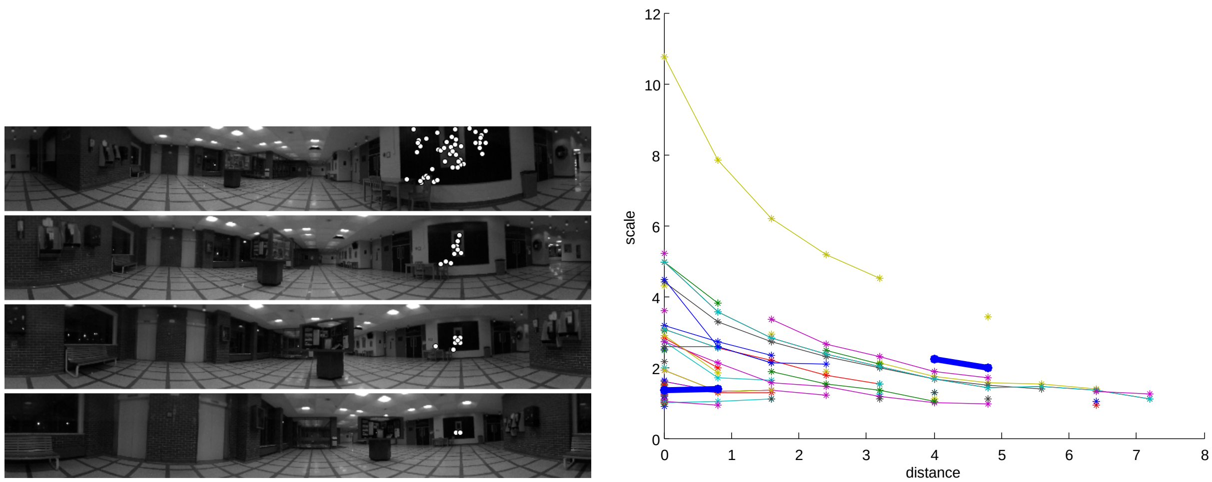
$$C^j = \{C^{j,x}, C^{j,y}, C^{j,\theta}, C^{j,\sigma}, \mathbf{C^{j,d}}\} \tag{1}$$

The feature's location within the image is $(C^{j,x}, C^{j,y})$, its orientation is $C^{j,\theta}$, its scale is $C^{j,\sigma}$, and its descriptor vector is $\mathbf{C^{j,d}}$.

As far as we are aware, Homing in Scale Space [1] was the first visual navigation method to make explicit use of $C^{j,\sigma}$ (henceforth referred to as σ if the context is clear). We have also recently employed σ to localize a robot along a trained route [36]. Informally, σ is the effective amount of Gaussian blurring required for a feature's distinctive characteristic to emerge (the distinctive characteristic being that the point is a local extrema with respect to both scale and space). Consider a landmark which yields one or more SIFT features. If the landmark is approached, it will take more blurring for the corresponding features to be detected. Thus, σ increases as the distance between the landmark and viewer decreases.

For our purposes we need only determine whether the distance to a landmark has increased or decreased with respect to a reference location. We utilize σ for this purpose. This substitution is valid as long as σ decreases monotonically as distance increases. Figure

3(a) shows a selection of panoramic images captured in the lobby of the S.J. Carew building at Memorial University. A total of 10 images were captured at increasing distances from a plaque on the wall. The top image shows the positions of SIFT features extracted from the vicinity of this plaque (features lying outside the large rectangular region surrounding the plaque were discarded). Subsequent images show the matched features for images at distances of 2.4, 4.8, and 7.2m from the top image. Figure 3(b) shows the scale σ of matched features versus distance from the reference location. A clear trend of decreasing scale with increasing distance is observable. Although, there are a few exceptions such as the feature indicated by the heavy trace.



(a) (b)

Figure 3: (a) Images taken from the lobby of the S.J. Carew building of Memorial University. Overlaid are the locations of features extracted from the vicinity of a plaque on the wall. (b) Plot of the relation between spatial distance and feature scale for the features extracted from the top image in (a). The heavy trace indicates one feature which exhibits an increase in scale with increasing distance, contrary to the general trend of decreasing scale with increasing distance.

Let S^j be the j^{th} SIFT feature extracted from the snapshot image and C^k be the k^{th} feature from the current image. If these features are matched, we can compute a quantity Δ_σ which indicates whether the feature has expanded or contracted.

$$\Delta_\sigma = S^{j,\sigma} - C^{k,\sigma} \tag{2}$$

If $\Delta_\sigma > 0$ then the feature has contracted. If $\Delta_\sigma < 0$ then the feature has expanded. A value of zero indicates no detectable change in apparent size.

Consider a matched feature which is generated by an object in the environment at position $\boldsymbol{f}$. Let $d(\boldsymbol{x}, \boldsymbol{f})$ represent the distance from a position $\boldsymbol{x}$ to the feature $\boldsymbol{f}$. The relationship between $d(\boldsymbol{x}, \boldsymbol{f})$ and σ is not straightforward. It depends upon the discretization of the scale-space pyramid, the relative positions of $\boldsymbol{x}$ and $\boldsymbol{f}$, and the physical size of the object that generates the feature. Nevertheless, we assume that when observing the same feature from two positions such as $\boldsymbol{c}$ and $\boldsymbol{s}$ that the following holds.

$$sign(\Delta_\sigma) = sign(d(\boldsymbol{c}, \boldsymbol{f}) - d(\boldsymbol{s}, \boldsymbol{f})) \tag{3}$$

The principles described below make use of this relationship, allowing us to compare the scale value of matched features and infer information about the sign of distance changes.

3.4. Principles

Homing in Scale Space is based on two simple principles:

1. Move towards features that have contracted ($\Delta_\sigma > 0$).
2. Move away from features that have expanded ($\Delta_\sigma < 0$).

To determine whether a feature has expanded or contracted, we compute a set of SIFT feature matches from S to C. Let $m^i = (S^j, C^k)$, represent the i^{th} matched pair. We determine whether a feature has contracted or expanded from the sign of Δ_σ as given in equation 2. If $\Delta_\sigma = 0$ then we exclude the feature pair, leaving a total of n matched pairs where the feature has either expanded or contracted from S to C. For each m^i we use the angular position of C^k to define a partial movement vector $\boldsymbol{v}^i$ which is a unit vector directed either towards the feature if it is contracted, or away from it if expanded (details in section 3.4.2). All partial movement vectors are added to produce an overall movement vector $\boldsymbol{h}$. The overall direction of movement α is then computed from $\boldsymbol{h}$.

$$\boldsymbol{h} = \frac{1}{|\sum_{i=0}^{n} \boldsymbol{v}^i|} \sum_{i=0}^{n} \boldsymbol{v}^i \tag{4}$$

$$\alpha = atan2(h_y, h_x) \tag{5}$$

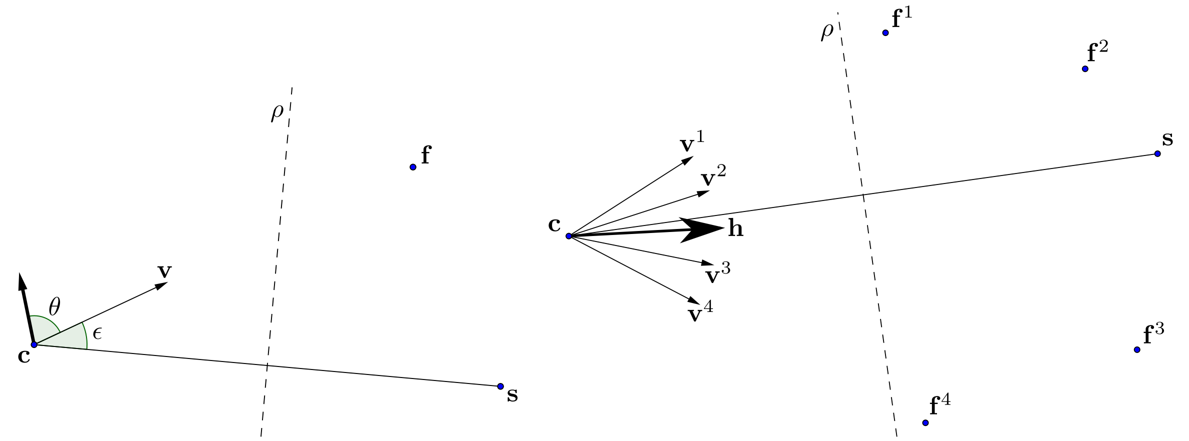
Notice that $\boldsymbol{h}$ is given as a unit vector, although this is not strictly necessary as we are only interested in its direction α.

3.4.1. Principle 1

Consider the case of a contracted feature as shown in figure 4(a). The robot's orientation at $\boldsymbol{c}$ is shown by the short thick vector. Feature $\boldsymbol{f}$ is seen at an angle θ with respect to the robot's orientation. This angle is sufficient to specify a unit vector $\boldsymbol{v}$ directed towards $\boldsymbol{f}$, which makes an angle ϵ with the line through $\overline{\boldsymbol{cs}}$. $|\epsilon|$ is the *angular error*, a value that would be zero in the ideal case ($\boldsymbol{f}$ co-linear with $\overline{\boldsymbol{cs}}$). Also, shown is ρ the perpendicular bisector of $\overline{\boldsymbol{cs}}$. As long as $\boldsymbol{f}$ lies on the same side of ρ as $\boldsymbol{s}$ then the distance from $\boldsymbol{c}$ to $\boldsymbol{f}$ will be greater than the distance from $\boldsymbol{s}$ to $\boldsymbol{f}$. Hence, the feature will appear to have contracted and Δ_σ should take on a positive value.

The unit vector $\boldsymbol{v}$ represents a partial motion vector corresponding to contracted feature $\boldsymbol{f}$. This vector would represent the ideal movement of the robot only if $\boldsymbol{f}$ was collinear with $\overline{\boldsymbol{cs}}$. If $\boldsymbol{f}$ lies on the snapshot side of ρ then ϵ is constrained to lie in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Further, $\epsilon = \pm\frac{\pi}{2}$ only if $\boldsymbol{f}$ lies directly on ρ at an infinite distance.

Thus, a movement towards a contracted feature at a finite distance yields a home vector with an angular error less than $\frac{\pi}{2}$. Franz et al argue that homing under this condition



(a) Single contracted feature

(b) Multiple contracted features

Figure 4: The perpendicular bisector of $\overline{\boldsymbol{cs}}$, denoted ρ, separates expanded from contracted features. Here features lie on the same side of ρ as $\boldsymbol{s}$ indicating contraction. (a) A single contracted feature is present. The partial movement vector $\boldsymbol{v}$ is directed towards this feature, which is at an error angle of ϵ from $\overline{\boldsymbol{cs}}$. (b) Four contracted features are present. $\boldsymbol{h}$ is the normalized sum of $\boldsymbol{v}^1$, $\boldsymbol{v}^2$, $\boldsymbol{v}^3$, and $\boldsymbol{v}^4$.

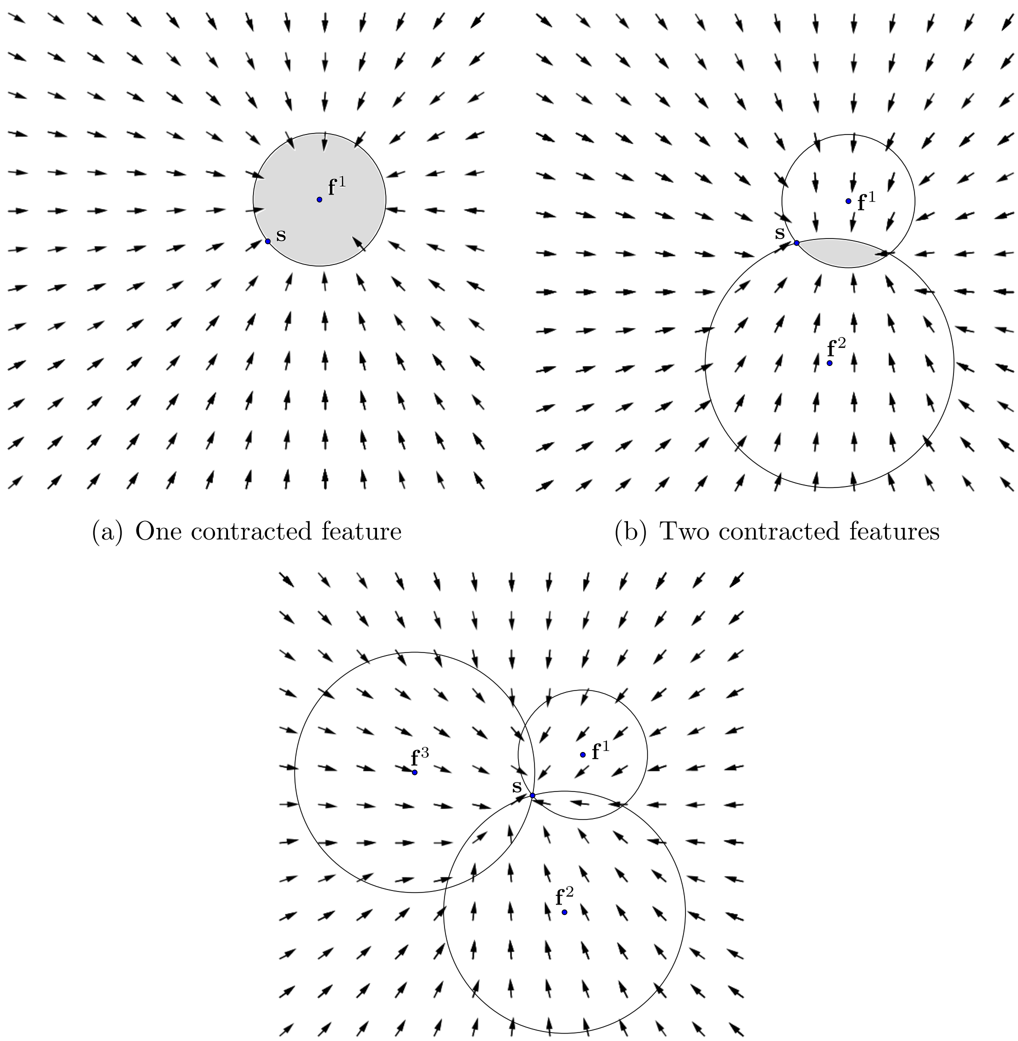
is convergent [19]. Let $d(\boldsymbol{c}(t), \boldsymbol{s})$ represent the distance from $\boldsymbol{c}(t)$ to $\boldsymbol{s}$ where $\boldsymbol{c}$ is now a function of time t. Since the angular error is always less than $\frac{\pi}{2}$ movements along $\boldsymbol{v}$ will yield a monotonic decrease in $d(\boldsymbol{c}(t), \boldsymbol{s})$. If guided by a single feature alone, the robot would reach a point at which the feature ceases to be a contracted feature. At this point, the following condition would hold,

$$d(\boldsymbol{c}(t), \boldsymbol{f}) = d(\boldsymbol{s}, \boldsymbol{f}). \tag{6}$$

This condition indicates that $\boldsymbol{c}(t)$ (i.e. the robot) lies on a circle centred at $\boldsymbol{f}$ that intersects $\boldsymbol{s}$. We will refer to this circle as the *scale horizon* for feature $\boldsymbol{f}$, so called because the feature's scale change Δ_σ will change from positive to negative as the circle is entered. The area enclosed by the scale horizon can be considered a *dead zone* with respect to the contracted feature. Homing for a single feature converges to this dead zone but goes no further. An example home vector field for a single feature is shown in figure 5(a).

When multiple contracted features are present, the scale horizons may have some degree of overlap. Only points that lie in the intersection of all scale horizons will belong to the dead zone. Thus, as more features are added the dead zone will tend to shrink and will typically disappear entirely after the addition of just a few features. Examples for two and three features are shown in figures 5(b) and 5(c). In the case of figure 5(c) three features are sufficient to eliminate the dead zone. In summary, principle 1 yields convergent homing to an area called the dead zone. When multiple contracted features are present the dead zone will typically disappear, yielding convergent homing to $\boldsymbol{s}$ from all points in the plane.

While convergence to the dead zone as $t \to \infty$ is an attractive property, we would prefer an angular error as close to 0 as possible to minimize the distance travelled. If we have an ensemble of contracted features, each denoted as $\boldsymbol{f}^i$, we can compute a unit movement



(a) One contracted feature

(b) Two contracted features

(c) Three contracted features

Figure 5: Home vectors produced by the application of equation 5 on contracted features only. A circle representing a feature's scale horizon surrounds each feature-generating object $\boldsymbol{f}^i$. Only regions within the intersection of all scale horizons lie in the dead zone. Such regions are shaded grey.

vector for each. If the angular distribution of features is approximately uniform, then the sum of all individual movement vectors $\boldsymbol{h}$ would point approximately towards $\boldsymbol{s}$ (see figure 4(b)). Yet even if the angular distribution of features was not uniform, the sum of individual movement vectors would still exhibit an error less than $\frac{\pi}{2}$, yielding convergent homing as described above.

3.4.2. Principle 2

Principle 2 is illustrated in figure 6. Features $\boldsymbol{f}^1$ and $\boldsymbol{f}^2$ lie on the same side of ρ as $\boldsymbol{c}$. Thus, they will appear to have expanded. The vectors $\boldsymbol{v}^1$ and $\boldsymbol{v}^2$ are now directed *away* from their corresponding features. For $\boldsymbol{v}^1$ this yields an angular error $\leq \frac{\pi}{2}$, but not for $\boldsymbol{v}^2$. The difference is that $\boldsymbol{f}^2$ lies in the region between the perpendicular bisector ρ and a parallel line through $\boldsymbol{c}$ called ρ'. We will call this region B (the bad region). Movement vectors for expanded features in region B exhibit angular error greater than $\frac{\pi}{2}$. Any expanded features not in region B will lie in region A. These features will yield convergent movement vectors since there is no dead zone associated with expanded features.

We can argue that region A will tend to be much larger than region B, and therefore will contain more features. If the distance $d(\boldsymbol{c}, \boldsymbol{s})$ is small relative to the size of the environment then this will likely be the case. If so, then the 'good' features in region A may outweigh the 'bad' features in region B. We have found this to be true in our experimental results. Also, if features are evenly distributed However, it must be acknowledged that convergent homing can not be guaranteed for principle 2.

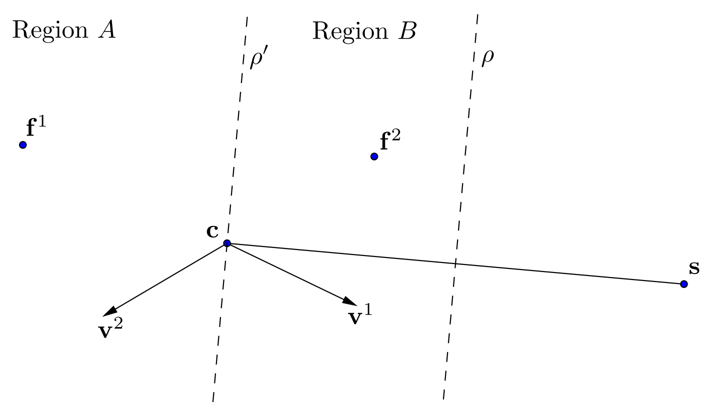


Figure 6: Feature-generating objects $\boldsymbol{f}^1$ and $\boldsymbol{f}^2$ lie on the same side of ρ as $\boldsymbol{c}$. Therefore, these features will have expanded in the current view image C and the corresponding partial movement vectors $\boldsymbol{v}^1$ and $\boldsymbol{v}^2$ point *away* from them. Regions A and B are defined with respect to ρ and ρ' as shown. The partial movement vector $\boldsymbol{v}^1$ for $\boldsymbol{f}^1$ in region A has an angular error less than $\frac{\pi}{2}$. However, the vector $\boldsymbol{v}^2$ for $\boldsymbol{f}^2$ has an error greater than $\frac{\pi}{2}$.

Implementation Details

We use panoramic images of our environment to represent views from the robot's perspective. These images are w pixels wide by h pixels high and represent a complete viewing

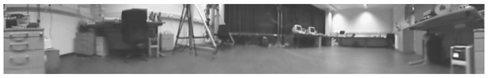
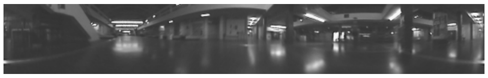
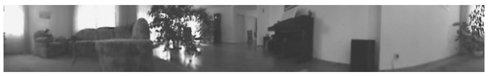
Sample Image	Name	Image Size	Capture Grid	Grid Spacing
	A1OriginalH	561×81	10×17	30cm
	CHall1H	561×81	10×20	50cm
	CHall2H	561×81	8×20	50cm
	Kitchen1H	583×81	12×9	10cm
	Moeller1H	583×81	22×11	10cm
	ISLab	346×50	9×8	61cm

Figure 7: Detailed information for each of the six image databases used.

angle of 2π in the horizontal direction and γ_{max} radians in the vertical direction. Each pixel represents a spacing of δ_x radians in azimuth, and δ_y radians in elevation, computable by:

$$\delta_x = \frac{2\pi}{w} \qquad \delta_y = \frac{\gamma_{max}}{h}$$

We therefore can convert a feature F^i with pixel coordinates $(F^{i,x}, F^{i,y})$ to angular coordinates (θ_i, γ_i).

$$\theta_i = \delta_x F^{i,x} \qquad \gamma_i = \delta_y F^{i,y}$$

For movements in the plane, only θ_i is required. We can compute a partial movement vector for feature F^i, which is directed towards contracted features but away from expanded features.

$$\boldsymbol{v}^i = \begin{cases} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} & \text{if } \Delta_\sigma > 0 \\ \\ \begin{bmatrix} \cos(\theta_i + \pi) \\ \sin(\theta_i + \pi) \end{bmatrix} & \text{if } \Delta_\sigma < 0 \end{cases} \tag{7}$$

Our method operates on pairs of features that have been matched from S to C. These matches are determined via the standard match criterion described by Lowe [32] in which a match is accepted only if it is significantly better than the second closest match.

4. Experimental Methods

4.1. Image Databases

Six image databases were used for testing. For each database a capture grid was defined on the floor of the capture area. Images were captured by a camera mounted upwards on a robot, viewing a hyperbolic mirror. Next, the image is projected onto a sphere. The final image is obtained by sampling the image for positions on the sphere taken at constant angular increments. This representation is convenient in that all pixels from a single image column correspond to the same azimuth, while all pixels from a single row correspond to the same elevation. Sample images of this format along with information on the image databases are found in figure 7.

The A1OriginalH, CHall1H, and CHall2H databases were captured at the University of Bielefeld. A1OriginalH was captured within the Robotics Lab of the Computer Engineering Group, while CHall1H and CHall2H are of the main hall of the university. Kitchen1H and Moeller1H were captured by Sven Kreft and Sebastian Ruwisch in a small kitchen and living room, respectively. All visible objects remained stationary throughout the collection process. More details on the collection of these databases can be found in [22] (covering A1OriginalH, CHall1H, and CHall2H) and [38] (covering Kitchen1H and Moeller1H). All of these databases have been made publicly accessible at `http://www.ti.uni-bielefeld.de/html/research/avardy/index.html`.

The ISLab database was captured at the Intelligent Systems laboratory at Memorial University. The setting for the database is a lab with an off white floor lit by fluorescent lighting. Since it is an active laboratory, some of the images contain people who move throughout the collection process. This active setting provides for a more challenging environment for homing to take place, since features occasionally vanish or change locations between images. The floor of the lab is tiled by square tiles which measure 30.5×30.5cm. Images were captured on a grid equal to every second tile spacing. The area surrounding the image capture can be seen in the floor plan depicted in figure 8. Additional details on the format of our images are available in [39].

In order to demonstrate the invariance of our method to rotation, input images will be rotated by a random amount before each test is performed. A circular shift of the image simulates the rotation of the robot about an axis perpendicular to the ground plane. Images are rotated by a randomly chosen angle θ_r in the range $[0, 2\pi)$. For some experiments we will also simulate a change in elevation of the robot by shifting the image upwards or downwards by a random amount $v_{\text{shift}} \in [0, h)$, where h is the image height. Unlike the horizontal shift induced by a rotation, vertical shifting will leave some portion of the image undefined. These undefined pixels will be filled in with black. See figure 9 for an example of an image that has been both rotated and vertically shifted. These vertical shifts allow us to test the robustness of our algorithm to changes in the position of the image horizon.

4.2. Configuration

We utilize David Lowe's SIFT implementation available from `http://www.cs.ubc.ca/~lowe/keypoints/`. Our method operates best when a large number of SIFT features have been extracted. We therefore modified several parameters in order to maximize feature production, while still maintaining accurate results. The values changed from those of Lowe's original implementation are as follows:

1. The number of scales at which keypoints are extracted is increased from 3 to 6 to increase the number of overall keypoints, while maintaining feasible running time.

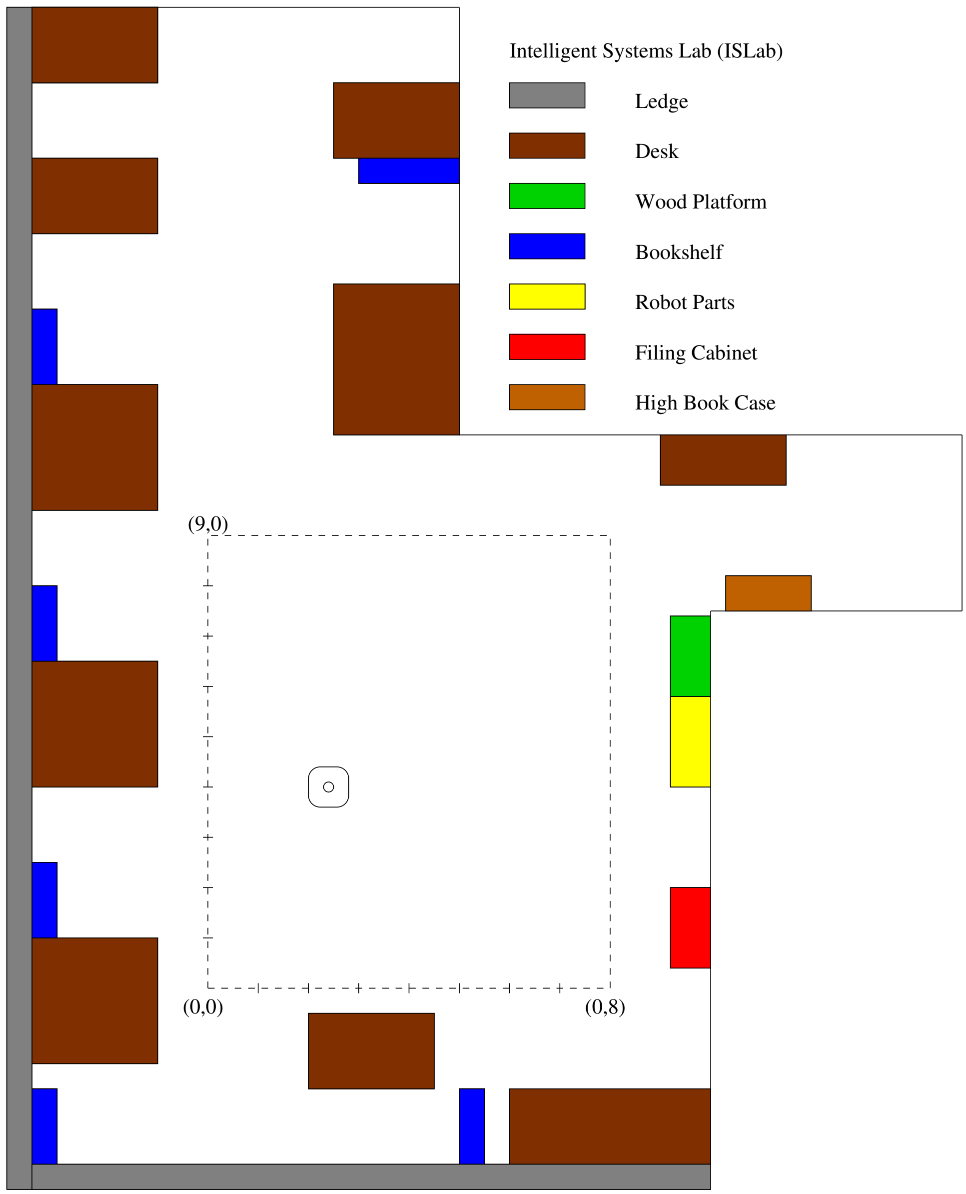


Figure 8: Diagram of the Intelligent Systems Lab at Memorial University of Newfoundland

Figure 9: Images from the A1OriginalH database taken at location (1,1). Top image shows the original image taken by the robot. Bottom image shows the image after a random amount of rotation, plus a random vertical shift. The remaining pixels after vertical shifting are filled in with black.

2. The peak threshold for the magnitude of the difference of Gaussian values is decreased from 0.08 to 0.01 in order to maintain more keypoints from areas of low contrast, since indoor environments often contain such areas.
3. The ratio of scores from best to second best feature match has been increased from 0.6 to 0.8. As discussed in [32], this change results in a marginal decrease in match accuracy while dramatically increasing the number of matches.

We use Ralf Möller's implementation of the warping method. Parameters for the warping method were selected to ensure fairness with respect to running time. We selected a discretization of 36 steps for all three movement parameters (α, ψ, ν). On an Intel Core2 2.13 GHz processor, this parameter selection resulted in an average execution time for the warping method which was 4.8% faster per snapshot than our method. We consider this to be a fair metric for results comparison.

4.3. Live Trial Implementation

Live robot trials were conducted using a Pioneer P3-AT robot. The environment for the live trials was exactly the same as described for the collection of the ISLab database. Five different snapshot positions were tested with the robot manually positioned at five different start positions at the start of each trial. Trials were terminated in the case of collisions with objects in the room or after 12 individual movements had been completed.

5. Results

5.1. Performance Metrics

Given two images S and C, the ideal visual homing algorithm computes α, the direction needed to move in order to reach $\boldsymbol{s}$ from $\boldsymbol{c}$. The robot will then move in the direction of α and determine whether or not it has arrived at the goal. In order to measure the accuracy of a given homing algorithm, we use two different performance metrics [22]. The first metric, angular error, is the difference between α and the true homing direction α_{ideal}.

The second metric is the return ratio, which measures the number of times the robot was able to successfully navigate to the goal location. Angular error results will be averaged over a set of start positions and describe only the error of instantaneous home vectors. The return ratio metric describes the overall success of a homing attempt. It is possible that in some pathological environments the majority of home vectors are accurate, except those close to the goal. The inaccurate home vectors close to the goal are in the minority but may have the effect of preventing the robot from reaching the goal for many start positions. In this case, the average angular error metric will indicate successful homing but the return ratio metric will indicate unsuccessful homing.

For results on the image databases, we have access to the true positions of both $\boldsymbol{s}$ and $\boldsymbol{c}$. Therefore, we can compute the ideal home angle as follows:

$$\alpha_{ideal}(\boldsymbol{s}, \boldsymbol{c}) \;=\; atan2(\boldsymbol{s}_y - \boldsymbol{c}_y, \boldsymbol{s}_x - \boldsymbol{c}_x) \tag{8}$$

thus, the angular error $AE(\boldsymbol{s}, \boldsymbol{c})$ can be found by:

$$AE(\boldsymbol{s}, \boldsymbol{c}) \;=\; \mathrm{diff}(\alpha_{ideal} - \alpha_{homing}) \tag{9}$$

where diff() is a function that yields the difference between two angles. We can then obtain an overall average angular error as follows:

$$AAE(\boldsymbol{s}) \;=\; \frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n} AE(\boldsymbol{s}, \boldsymbol{c}_{xy}). \tag{10}$$

where $AE(\boldsymbol{s}, \boldsymbol{s}) = 0$.

To obtain a measure of performance for the entire image database we can define the overall average angular error $OAAE(\boldsymbol{db})$, which computes the overall average of AAE for all snapshot images in database $\boldsymbol{db}$.

$$OAAE(\boldsymbol{db}) \;=\; \frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n} AAE(\boldsymbol{s}_{xy}). \tag{11}$$

The second performance metric is the return ratio. The return ratio is computed by carrying out simulated homing trials on the capture grid of a particular database. We declare a trial to be successful if the simulated robot was able to return to within a given distance threshold of $\boldsymbol{s}$. We define $\mathrm{ret}(\boldsymbol{c}, \boldsymbol{s})$ as a binary valued function with a value of 1 for successful homing and 0 for unsuccessful homing. $\mathrm{ret}(\boldsymbol{c}, \boldsymbol{s})$ is evaluated on image database $\boldsymbol{db}$ as follows:

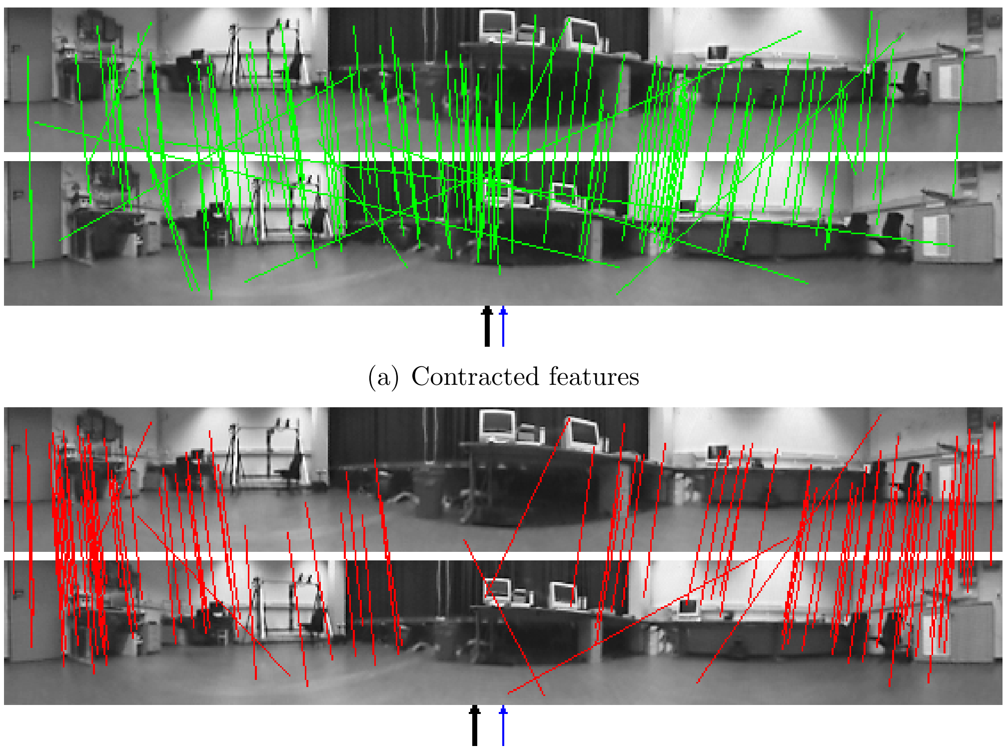
1. For positions $\boldsymbol{c}$ and $\boldsymbol{s}$ apply the homing algorithm on C and S to obtain α_{homing}.
2. Calculate the new position of the simulated robot by moving in the direction of α_{homing}: $\boldsymbol{c}_{new} = (\boldsymbol{c}_x + \mathrm{round}(\cos(\alpha_{homing})), \boldsymbol{c}_y + \mathrm{round}(\sin(\alpha_{homing})))$.
3. If $\boldsymbol{c}_{new} = \boldsymbol{s}$, homing is successful. If $\boldsymbol{c}_{new}$ is outside the boundary determined by the capture grid, or is the same as a previously visited $\boldsymbol{c}$ (loop), then the trial is considered unsuccessful. Otherwise, return to step 1 with $\boldsymbol{c} = \boldsymbol{c}_{new}$.

If we iterate this process over all possible $\boldsymbol{c}$ for all possible $\boldsymbol{s}$, we can determine the total return ratio $TRR(\boldsymbol{db})$ as the percentage of homing trials that succeed.

$$RR(\boldsymbol{s}) = \sum_{x=1}^{m}\sum_{y=1}^{n} \mathrm{ret}(\boldsymbol{c}_{xy}, \boldsymbol{s}_{xy})/mn \tag{12}$$

$$TRR(\boldsymbol{db}) = \sum_{x=1}^{m}\sum_{y=1}^{n} RR(\boldsymbol{s}_{xy})/mn. \tag{13}$$

Figure 10 shows an example of the operation of our algorithm. In this case the two images are taken from the same orientation for ease of interpretation.



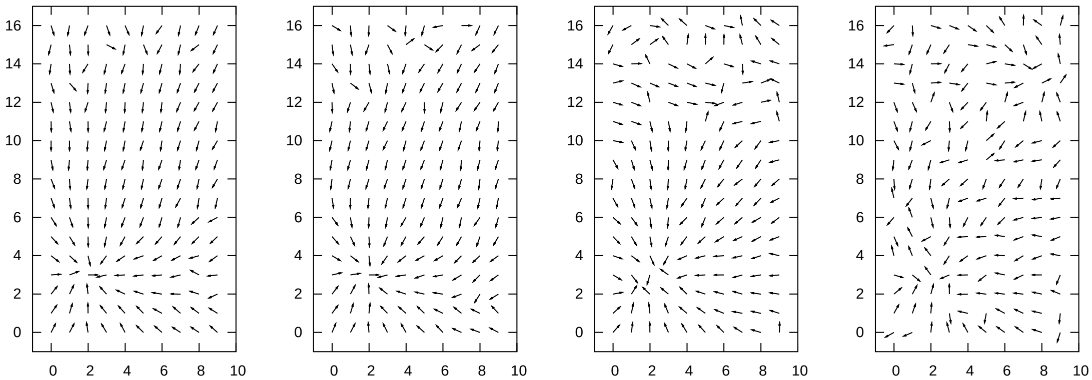
(a) Contracted features

(b) Expanded features

Figure 10: Example of the application of HiSS. For both (a) and (b) the snapshot image shown on top is from position $(5, 8)$ of A1OriginalH. The current image below is from position $(3, 8)$. Lines between the images indicate matches between SIFT keypoints. In (a) these lines connect the contracted features in C with their matches in S. In (b) expanded features are shown. The thin arrow indicates the true home position while the thicker arrow indicates the computed home direction by contracted (a) or expanded (b) features.

5.2. Notation

Each test was performed using both homing methods. Wherever 'H' or 'HiSS' is noted in a legend or table, it represents the results for the homing in scale space method. Wherever



(a) HiSS, rot. (b) HiSS, rot. + v. shift (c) Warping, rot. (d) Warping, rot. + v. shift

Figure 11: Homing vector images with $\boldsymbol{s}$ set to $(2, 3)$. The two plots on the left show the application of HiSS with random rotation (first, AAE=12.3°) and combined vertical shift (second, AAE=18.1°). The two plots on the right show the warping method with random rotation (third, AAE=39.2°) and combined vertical shift (fourth, AAE=59.4°)

'W' or 'Warp' is noted in a legend or table, it represents the results for the warping method. Since all tests were done with a certain level of vertical shifting, wherever 0px, 5px, 15px, or 24px is noted, it corresponds to the maximum random vertical shift for that particular trial. For example, '15H' or 'HiSS15' both refer to a trial performed by the homing in scale space method under uniform$(0, 15)$ pixel vertical shift, where uniform(a, b) returns a uniformly-distributed random number in the range $[a, b]$.

5.3. Results on Image Databases

In figure 11 we see the results of homing to location $(2, 3)$ in the A1OriginalH database from every other location in the database. Computed homing angles are represented by unit vectors.

Figures 12 and 13 are of grayscale grids plotted for each (x, y) location within each database. The gray scale value for a particular location within a database is scaled from black (0) to white (maximum of $max(OAAE(hiss), OAAE(warping))$ for a particular database). This view allows us to see which locations in a particular environment perform well (darker), or poorly (lighter). Note that the aspect ratio for these figures is not 1:1, refer to the axes for coordinate information. These results are summarized for the angular error metric in figure 14 and for the return ratio metric in figure 15.

From figures 12, 13, and 14 it appears that the OAAE for our method is lower than that for the warping method. However, we must show that there is indeed a statistically significant difference between the two methods. In order to determine which tests to perform, we first analyzed the distribution of our data. For the angular error data, we used the Shapiro-Wilk, or W normality test [40, 41]. Upon running the W test for each of the data sets individually, as well as all combined data sets as a whole, each test returned a result of $p < 2.2e - 16$, indicating that our data is not normally distributed. For this reason, we used the sign

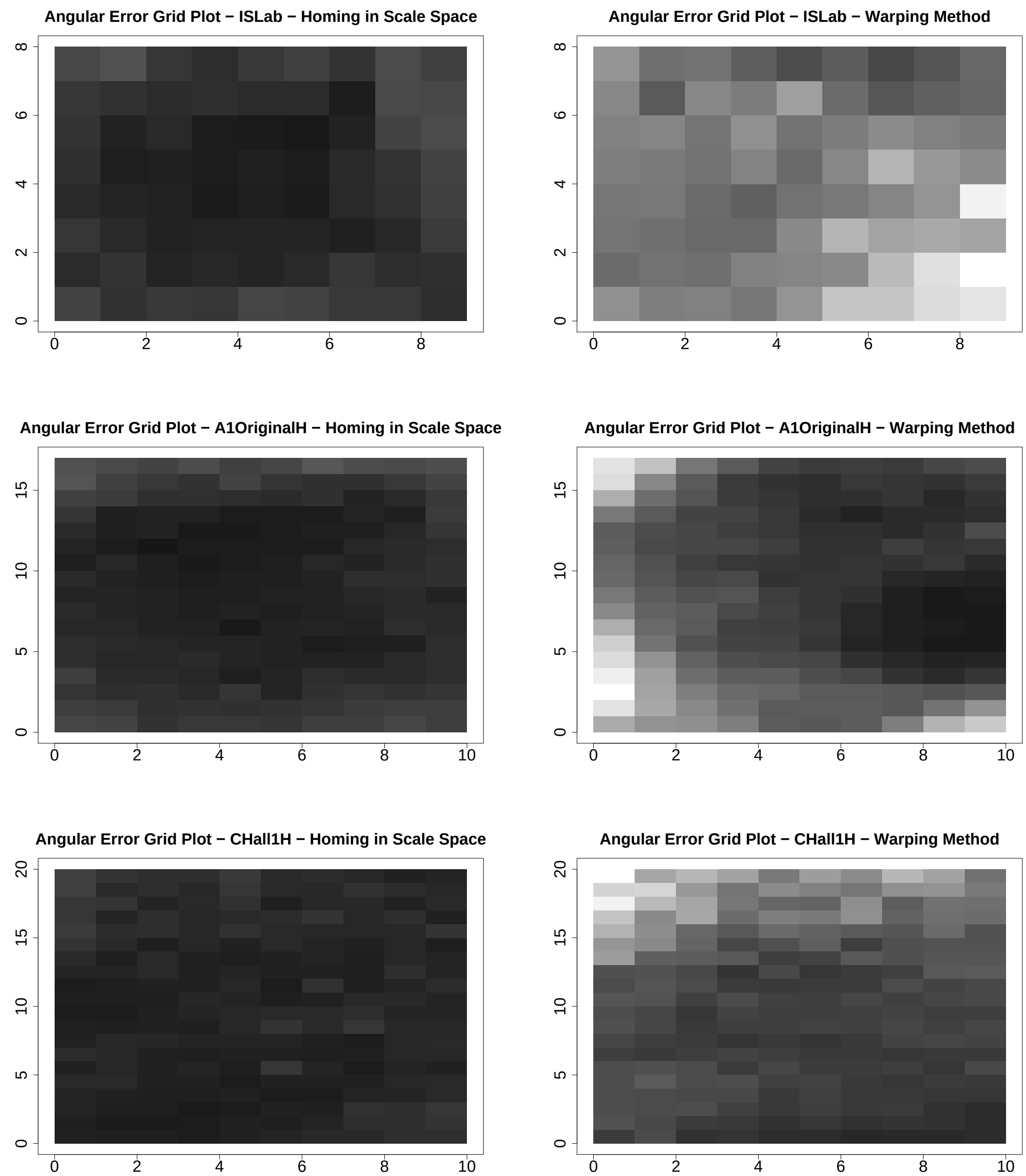


Figure 12: Grids showing AAE results for ISLab, A1OriginalH, and CHall1H databases.

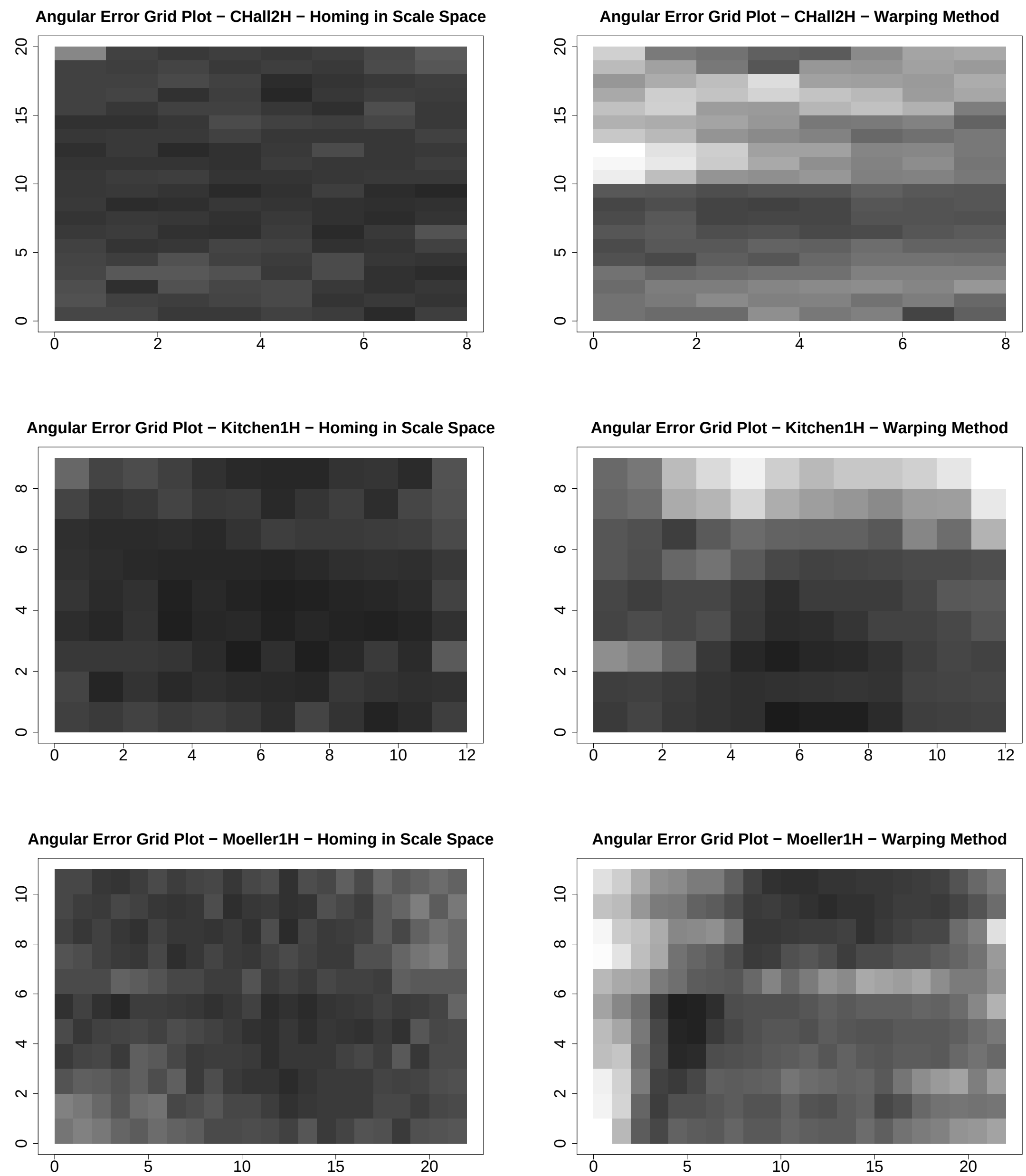
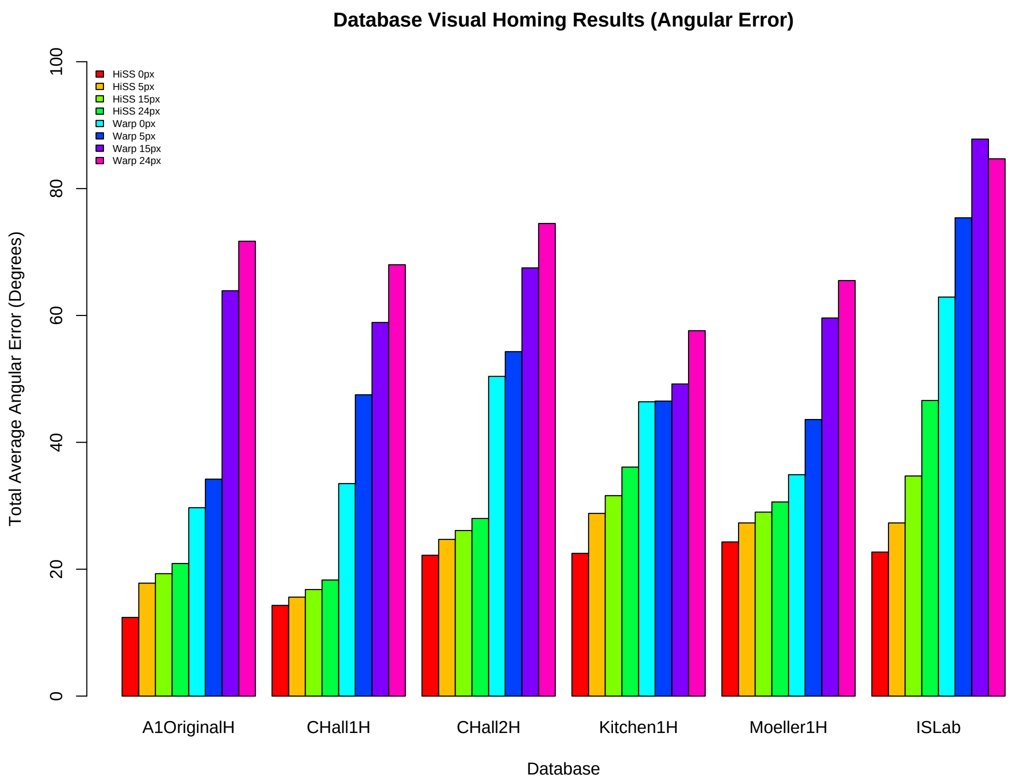
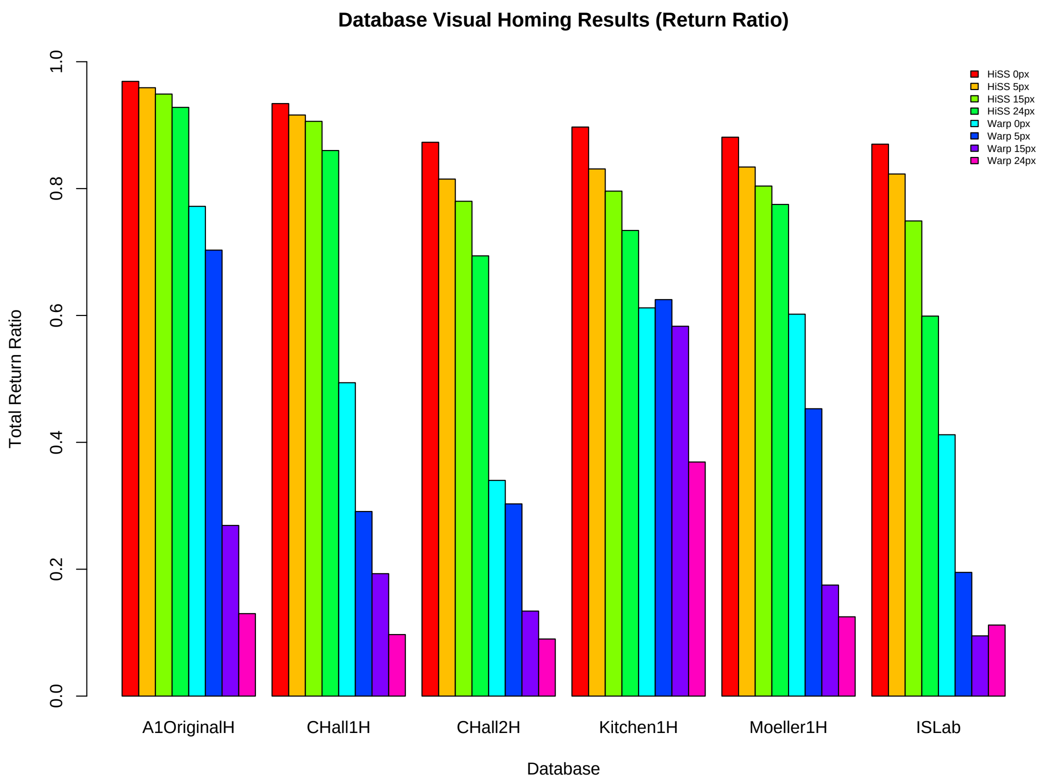


Figure 13: Grids showing AAE results for CHall2H, Kitchen1H, and Moeller1H databases.



Database	0H	5H	15H	24H	0W	5W	15W	24W
A1originalH	12.4°	17.8°	19.3°	20.9°	29.7°	34.2°	63.9°	71.7°
Chall1H	14.3°	15.6°	16.8°	18.3°	33.5°	47.5°	58.9°	68.0°
Chall2H	22.2°	24.7°	26.1°	28.0°	50.4°	54.3°	67.5°	74.5°
Kitchen1H	22.5°	28.8°	31.6°	36.1°	46.4°	46.5°	49.2°	57.6°
Moeller1H	24.3°	27.3°	29.0°	30.6°	34.9°	43.6°	59.6°	65.5°
RobISLab	22.7°	27.3°	34.7°	46.6°	62.9°	75.4°	87.8°	84.7°

Figure 14: Database Results - Angular Error



Database	0H	5H	15H	24H	0W	5W	15W	24W
A1originalH	0.969	0.959	0.949	0.928	0.772	0.703	0.269	0.130
Chall1H	0.934	0.916	0.906	0.860	0.494	0.291	0.193	0.097
Chall2H	0.873	0.815	0.780	0.694	0.340	0.303	0.134	0.090
Kitchen1H	0.897	0.831	0.796	0.734	0.612	0.625	0.583	0.369
Moeller1H	0.881	0.834	0.804	0.775	0.602	0.453	0.175	0.125
RobISLab	0.870	0.823	0.749	0.599	0.412	0.195	0.095	0.112

Figure 15: Database Results - Return Ratio

test which is applicable even if the data is not normally distributed [42]. The alternative hypothesis tested is that $AE(hiss) - AE(warp) < 0$, representing superior performance of HiSS over warping. A P-value < 0.05 is sufficient to support this alternative hypothesis [42, 43, 44]. The results from these tests are shown in figure 16. In all cases the P-value is < 0.05 indicating superior performance of HiSS over Warping.

5.4. Distance Estimation

As mentioned in section 3.2 visual homing can be achieved by incremental movements in the direction α. However, we can reach the goal more efficiently using some estimate of the distance r.

In [39] we considered a variety of image and feature-based measures in order to arrive at a quantity with a consistently high correlation with r. The best measure found was the percentage of SIFT keypoints matched, which we denote $M_{\%}$. Figure 17 presents plots of $M_{\%}$ versus the true distance r for all image databases. The relationship between these quantities appears to be exponential in nature:

$$r = ae^{bM_{\%}} \tag{14}$$

We used nonlinear regression using the R stats package to find the best parameters a and b for each image database. The results are overlaid on the raw data in figure 17. The overlaid graphs show 4 functions (one for each of the 0, 5, 15, 24 pixel vertical shifts). Note that due to the similarity of the resulting values of a and b the lines are difficult to distinguish. This indicates that the estimated relationships between $M_{\%}$ and r are relatively resistant to vertical shift. Figure 18 provides tables of the computed values of a and b.

We can see by figures 17 and 18 that the distance estimation function fits nicely to the exponential curve. The function also remains remarkably similar despite large vertical shifting within the image (represented by the different lines), making this method for distance estimation feasible for environments without level movement surfaces. One downside to this approach however is that as the true distance from the goal increases, so does the error in the function. At areas in the graph where the slope of the computed function has a larger magnitude, similar values of $M_{\%}$ can yield dramatically different distances. This would lead us to believe that this distance estimation method will be less accurate for long-range homing, but become more accurate as we approach the goal.

Another issue of note is the fact that this function varies with image dimensions. Homing within an environment using images with a height of 50 pixels will yield a different distance estimation function than an image with a height of 100 pixels. Experimentally, we have found that as resolution increases, more keypoints are found, and a higher value for $M_{\%}$ results.

5.5. ISLab Trials

To test our algorithm on our live robot, we used the environment of the ISLab database. Five different goal locations were chosen, with 5 starting locations for each goal location spaced evenly throughout the environment. The robot takes an image at its current location, Sign Test (With Alt. Hyp. HiSS-Warping < 0) - No Pixel Vertical Shift

Database	Samples	Mean	Median	95% CI	S-Value	P-Value
A1originalH	28900	-0.236	-0.054	$(-\pi, -0.051)$	11873	2.2e-16
Chall1H	40000	-0.318	-0.120	$(-\pi, -0.116)$	14252	2.2e-16
Chall2H	25600	-0.471	-0.255	$(-\pi, -0.246)$	8571	2.2e-16
Kitchen1H	11664	-0.375	-0.111	$(-\pi, -0.102)$	4497	2.2e-16
Moeller1H	58564	-0.197	-0.003	$(-\pi, 0.0)$	28851	0.0057
RobISLab	5184	-0.707	-0.429	$(-\pi, -0.399)$	1287	2.2e-16

Sign Test (With Alt. Hyp. HiSS-Warping < 0) - 5 Pixel Vertical Shift

Database	Samples	Mean	Median	95% CI	S-Value	P-Value
A1originalH	28900	-0.287	-0.069	$(-\pi, -0.066)$	11580	2.2e-16
Chall1H	40000	-0.556	-0.251	$(-\pi, -0.244)$	11481	2.2e-16
Chall2H	25600	-0.517	-0.316	$(-\pi, -0.305)$	7991	2.2e-16
Kitchen1H	11664	-0.309	-0.084	$(-\pi, -0.074)$	4859	2.2e-16
Moeller1H	58564	-0.285	-0.052	$(-\pi, -0.049)$	26075	2.2e-16
RobISLab	5184	-0.841	-0.659	$(-\pi, -0.621)$	1140	2.2e-16

Sign Test (With Alt. Hyp. HiSS-Warping < 0) - 15 Pixel Vertical Shift

Database	Samples	Mean	Median	95% CI	S-Value	P-Value
A1originalH	28900	-0.778	-0.528	$(-\pi, -0.513)$	6654	2.2e-16
Chall1H	40000	-0.734	-0.409	$(-\pi, -0.399)$	9888	2.2e-16
Chall2H	25600	-0.724	-0.541	$(-\pi, -0.525)$	6702	2.2e-16
Kitchen1H	11664	-0.307	-0.079	$(-\pi, -0.067)$	5016	2.2e-16
Moeller1H	58564	-0.535	-0.243	$(-\pi, -0.234)$	20712	2.2e-16
ISLab	5184	-0.927	-0.915	$(-\pi, -0.874)$	1133	2.2e-16

Sign Test (With Alt. Hyp. HiSS-Warping < 0) - 24 Pixel Vertical Shift

Database	Samples	Mean	Median	95% CI	S-Value	P-Value
A1originalH	28900	-0.885	-0.718	$(-\pi, -0.703)$	5903	2.2e-16
Chall1H	40000	-0.867	-0.638	$(-\pi, -0.625)$	8217	2.2e-16
Chall2H	25600	-0.812	-0.697	$(-\pi, -0.683)$	6057	2.2e-16
Kitchen1H	11664	-0.375	-0.133	$(-\pi, -0.120)$	4796	2.2e-16
Moeller1H	58564	-0.609	-0.386	$(-\pi, -0.376)$	18772	2.2e-16
ISLab	5184	-0.665	-0.606	$(-\pi, -0.573)$	1512	2.2e-16

Figure 16: Tables representing the results from the sign test applied to angular error data for HiSS-Warping.

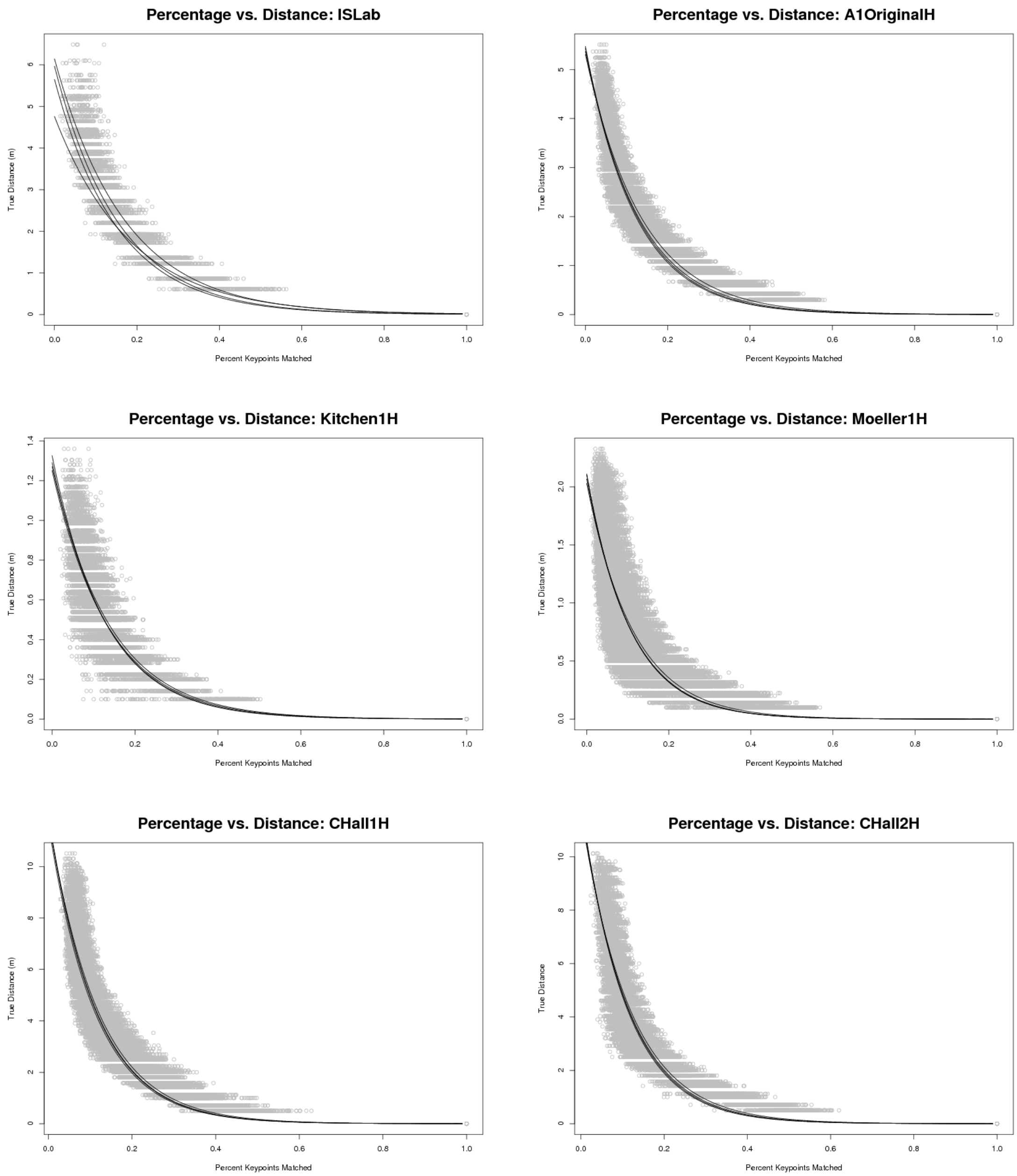


Figure 17: Percentage Matched vs. Distance graphs for each database.

Database	Trial	a	b	a std. err	b std. err	RSE
Islab	0px Vert	10.06780	-5.85665	0.05436	0.04763	0.8335
	5px Vert	9.78177	-6.43592	0.06124	0.06236	0.9595
	15px Vert	9.26169	-6.53532	0.06932	0.07979	1.144
	24px Vert	7.8007	-5.4061	0.0731	0.1028	1.494
A1OriginalH	0px Vert	17.68985	-7.27702	0.03602	0.02122	1.24
	5px Vert	17.89868	-7.73030	0.03793	0.02307	1.269
	15px Vert	18.23209	-8.03045	0.04010	0.02428	1.286
	24px Vert	18.11404	-8.17131	0.04154	0.02585	1.344
CHall1H	0px Vert	23.85965	-8.45522	0.05915	0.02441	1.668
	5px Vert	23.85637	-8.75030	0.06172	0.02625	1.734
	15px Vert	23.82518	-8.82529	0.06291	0.02698	1.772
	24px Vert	23.32249	-8.90887	0.06561	0.02945	1.907
CHall2H	0px Vert	23.37360	-8.50571	0.08667	0.03625	1.88
	5px Vert	23.71199	-8.87797	0.09259	0.03905	1.941
	15px Vert	24.10059	-9.11256	0.10206	0.04246	2.037
	24px Vert	24.07061	-9.26291	0.10868	0.04574	2.144
Kitchen1H	0px Vert	12.71268	-7.15042	0.07095	0.05730	1.447
	5px Vert	12.91063	-7.58406	0.07962	0.06509	1.53
	15px Vert	13.26301	-7.77610	0.08449	0.06660	1.538
	24px Vert	12.50825	-7.31220	0.09264	0.07631	1.76
Moeller1H	0px Vert	20.28980	-8.72208	0.05359	0.03502	2.685
	5px Vert	20.66887	-9.24524	0.05959	0.03914	2.787
	15px Vert	21.09868	-9.46355	0.06302	0.04028	2.808
	24px Vert	21.01630	-9.41256	0.06539	0.04160	2.886

Figure 18: Table of results for functions plotted in figure 17. a and b correspond to the values output by performing non linear regression on function $r = ae^{bM_{\%}}$. Standard errors for a and b, as well as the residual standard error (RSE) are also included.

compares it to the goal image, computes the estimated values for r and α, turns in the direction of α and moves a distance of $r/2$. Moving the full distance r on each step can lead the robot to overshoot the goal and then oscillate around it. We found that moving a distance of $r/2$ yields more stable behaviour. This process repeats until the robot believes it is within 30cm of the goal (success) or for a maximum of 12 iterations (failure). Values for r were computed from the fitted exponential function of $M_{\%}$ discussed above on the ISLab database. A real-time distance estimator is discussed in the future work section.

It was our original intention to compare homing in scale space to the warping method live on the robot. However, the warping method was found to be too inaccurate to carry out the trials. Of several dozen initial tests, the robot would inevitably veer off the allotted limits for navigation. We suspect this is due to the nature of the images captured by the robot. Due to unevenness in the floor and slight discrepancies in the diameter of the robot's wheels, both the height and inclination of the robot's camera varied slightly as it travelled across the floor. Since the warping method relies heavily on the stability of the horizon within an image, we believe that this variance caused enough shift of the image horizon to cause the warping method to perform poorly. Due to this, results for live trials using the warping method are not included.

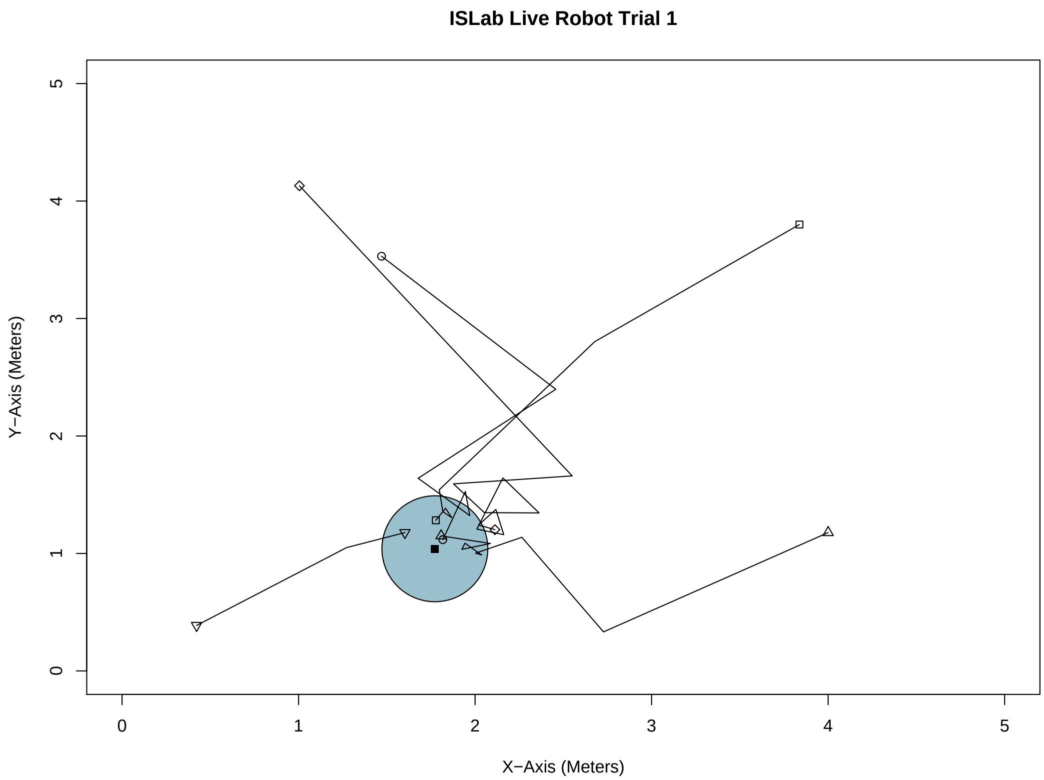
We will define two types of success for our live robot trials. Type A success means the robot came to stop within both an estimated distance of 30cm and an actual distance of 30cm. Type B success means that at some point the robot came within a true distance of 30cm of the goal, but did not stop due to error in its distance estimation. If the robot passed within 30cm of the goal at any point during a trial, but estimated it was not within the threshold, we record it as having been an undetected arrival (UA). Therefore, type B success is equivalent to any trial which recorded an undetected arrival without achieving type A success. Figures 20 through 24 show results for each goal position, along with a table of the associated estimated distance, actual distance, and distance estimate error for the final step of each homing trial.

For the 25 homing trials conducted, 21 resulted in type A success and 4 resulted in type B success. 14 of the trials resulted in the recording of an undetected arrival, which means that the method is actually getting closer to the goal than its distance estimation function would lead us to believe.

This effect is illustrated in figure 25 where we plot the relationship between the actual distance from the goal r_a and the error:

$$r_{err} = |r - r_a| \tag{15}$$

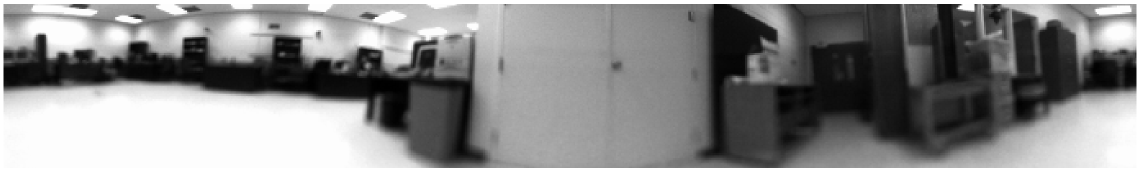
As r_a increases, so does r_{err}. Using the Spearman method of correlation between these two values yields a correlation coefficient of 0.784, which strongly reinforces this relationship. The second graph is a histogram of r_{err}, showing a possible reason for the high number of UAs in the live trials. The distance estimation function nearly always returns a value which is higher than that of the actual distance to the goal, with a mean of 0.462m and a median of 0.195cm. A possible reason for this is the fact that the distance estimation function was computed from the ISLab database, in which images were spaced 61cm apart. Since the

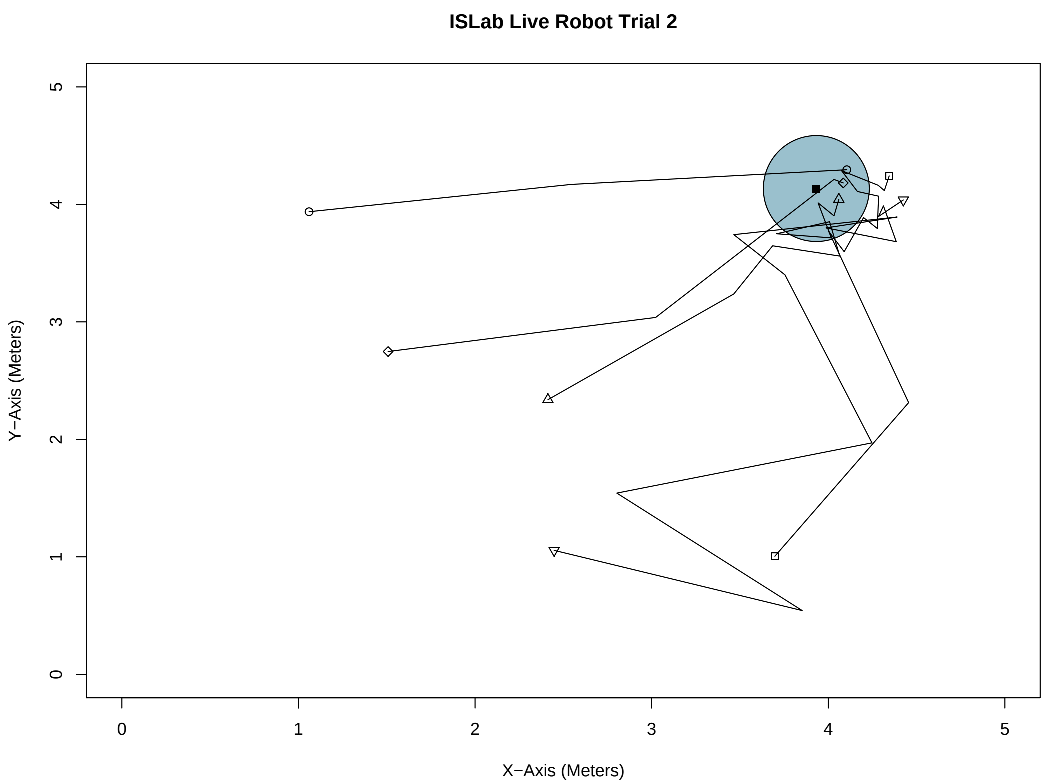


Trial		Est Dist	Act Dist	Error	Success	UA	Steps
1	○	0.25	0.09	0.16	A	NO	5
	□	0.3	0.24	0.06	A	YES	6
	◇	0.43	0.38	0.05	B	YES	12
	△	0.26	0.12	0.14	A	YES	7
	▽	0.28	0.22	0.06	A	NO	2

Figure 19: ISLab Live Homing Trial 1

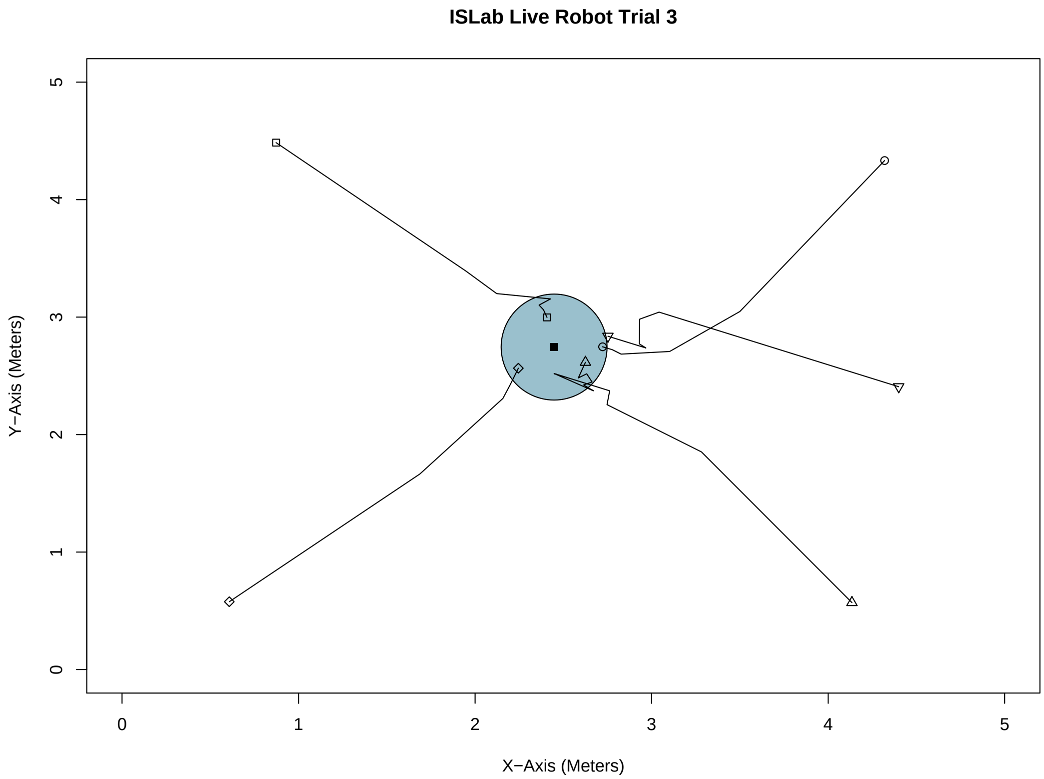
Figure 20: ISLab live homing trial 1. The plot above shows the positions of the robot as it approaches the goal area which is indicated by the shaded circle. The table below gives information on the final robot position for the corresponding homing attempt. Distance and error units in the table are given in metres.





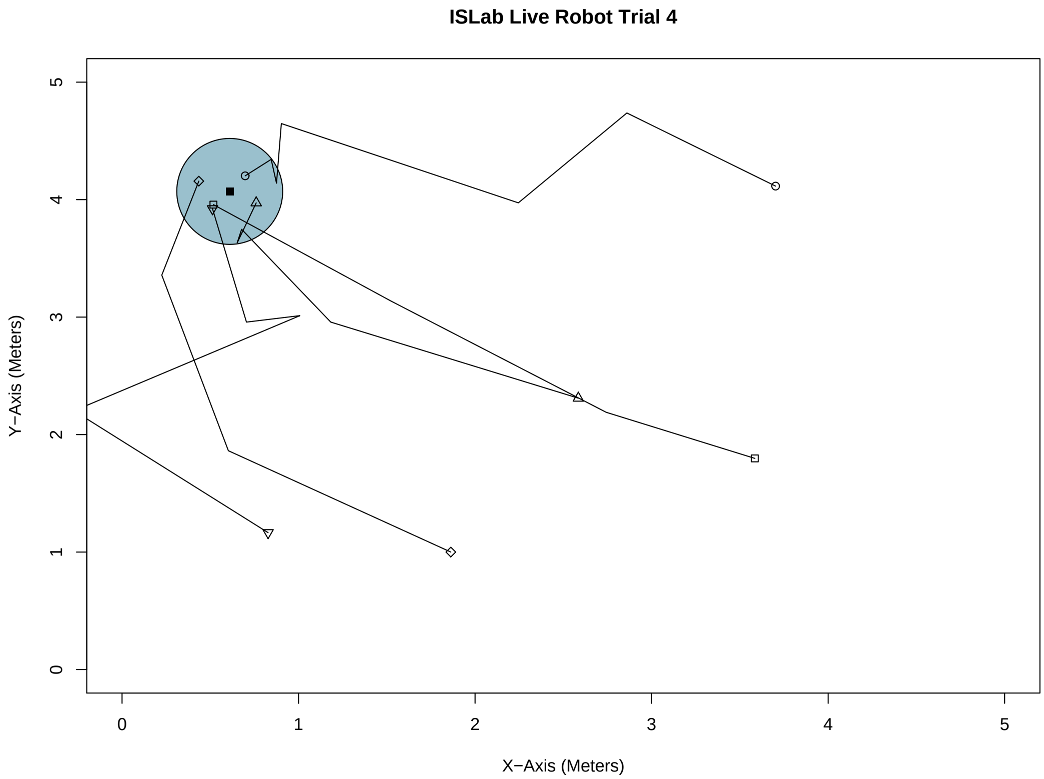
Trial		Est Dist	Act Dist	Error	Success	UA	Steps
2	○	0.29	0.24	0.06	A	NO	2
	□	0.57	0.43	0.15	A	YES	12
	◇	0.23	0.16	0.07	A	YES	3
	△	0.29	0.16	0.13	A	YES	9
	▽	0.62	0.50	0.12	B	YES	12

Figure 21: ISLab live homing trial 2.



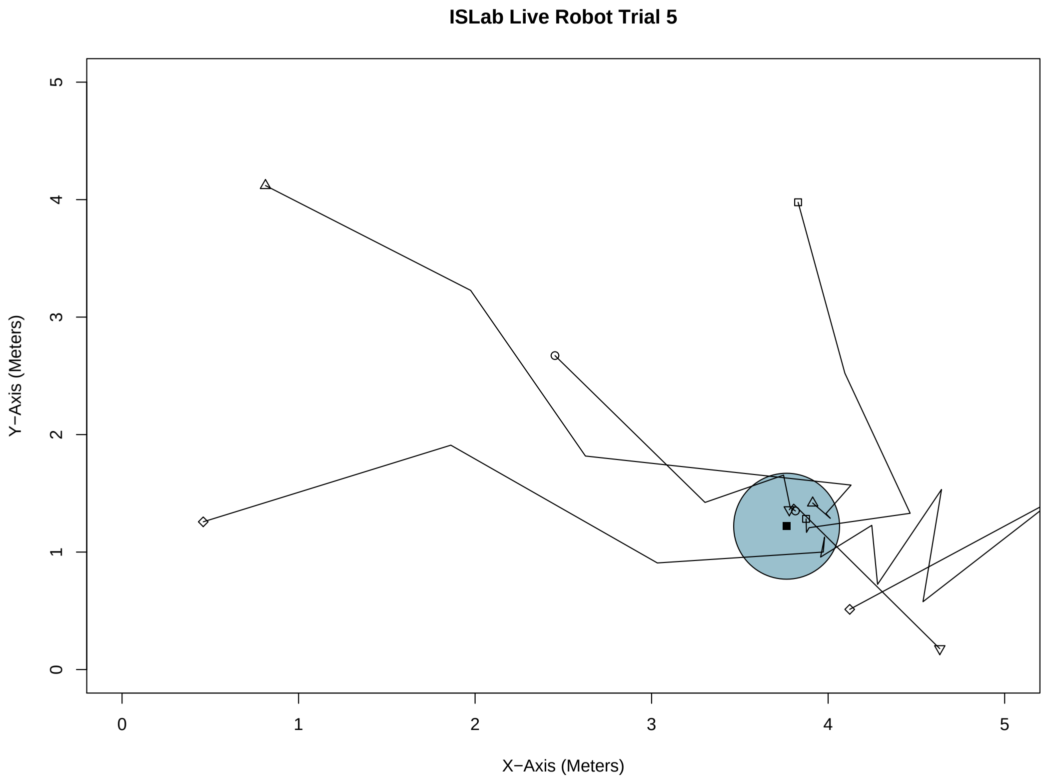
Trial		Est Dist	Act Dist	Error	Success	UA	Steps
3	○	0.29	0.27	0.02	A	NO	5
	□	0.28	0.26	0.02	A	NO	6
	◇	0.27	0.27	0.00	A	NO	4
	△	0.29	0.22	0.07	A	YES	10
	▽	0.29	0.32	0.03	A	NO	5

Figure 22: ISLab live homing trial 3.



Trial		Est Dist	Act Dist	Error	Success	UA	Steps
4	○	0.27	0.16	0.12	A	YES	6
	□	0.27	0.15	0.12	A	NO	3
	◊	0.24	0.20	0.05	A	NO	3
	△	0.25	0.18	0.07	A	NO	4
	▽	0.27	0.18	0.09	A	NO	4

Figure 23: ISLab live homing trial 4.



Trial		Est Dist	Act Dist	Error	Success	UA	Steps
5	○	0.28	0.14	0.14	A	YES	4
	□	0.26	0.13	0.13	A	YES	5
	◊	N/A	0.79	N/A	B	YES	12
	△	0.25	0.25	0.00	A	YES	6
	▽	0.29	0.14	0.15	A	YES	2

Figure 24: ISLab live homing trial 5.

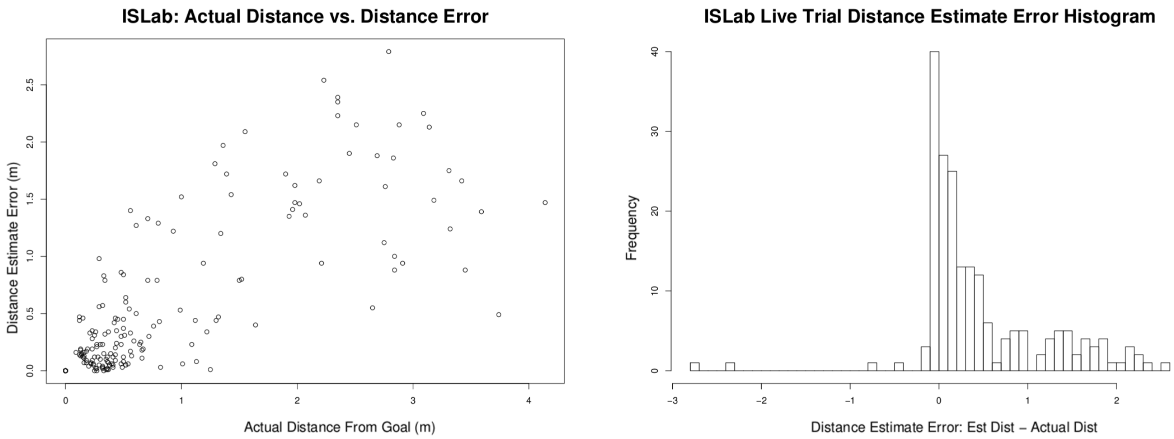


Figure 25: Graph (left) of actual distance from goal r_a vs. distance error $r_{err} = |r - r_a|$, along with the error histogram (right).

main purpose of the distance estimation function is to detect close proximity to the goal, it would be preferable to estimate this function with finer-grained resolution—particularly for smaller distance values.

6. Discussion

Our tests have demonstrated the superior performance of our method over the warping method for all six image databases. Homing in scale space yielded a dramatically lower angular error, as well as a higher return ratio than the warping method. The random horizontal rotations and vertical shifts that were incorporated into the database experiments were included to demonstrate our method's invariance to orientation changes and robustness to vertical image shifts.

Results from the live robot trials were in agreement with those from the image databases. The type A success rate was found to be 84%. If we combine this with type B successes, we see that homing in scale space was able to bring the robot to within 30cm of the goal in all cases. These results were obtained in an environment where the warping method was unable to achieve any measurable success.

6.1. Future Work

Recall the value of Δ_σ which was used to determine whether a feature was classified as contracted or expanded. In the case of images captured at nearby locations, we could see many very small values for Δ_σ. In the presence of camera noise and improper focus, the chance of misclassification between contracted and expanded features may be high. We experimented with a threshold parameter for filtering matches with low values of Δ_σ. However, the results were inconsistent across different image databases. A more sophisticated classification strategy should be investigated in the future.

Distance estimation is another area where improvements can be made. The distance estimation formula used in our live robot trials was computed using nonlinear regression based on data from the ISLab database. We propose that a distance estimation function for a particular environment could be calculated using relative motion data collected by an inertial measurement unit (IMU), thus eliminating the need for an existing database. Assuming that the robot captured the goal image and then moved away from it (e.g. in the context of learning a route or topological map of the environment) we could estimate the true distance to the goal via the IMU. The relationship between distance and $M_{\%}$ could then be learned on-line.

In this paper we have shown that homing in scale space is invariant to rotations of an image about an axis perpendicular to the ground plane. However, we would like to demonstrate more conclusively that the algorithm is invariant to any 3D rotation. We have captured a database of images taken from a variety of roll, pitch, and yaw angles. However, since we are using the same camera system as in this paper, the images are not truly omnidirectional. Limitations in the field-of-view have an impact on the algorithm's performance and we are still determining the best way of analyzing these results. It would be interesting to extend our technique for application on unmanned aerial vehicles (UAVs). Since aerial vehicles travel in 3D we would need to augment the algorithm by computing both the angle of azimuth (i.e. α) and the angle of elevation. This change could easily be accommodated by making the partial movement vectors defined in equation 7 three-dimensional.

Visual homing techniques can be applied only when the robot lies within the catchment area of the goal location. Our image database results indicate a catchment area covering the entire capture grid (areas ranged from 1.08 to 50 m^2). Nevertheless, other techniques will certainly be required to guide the robot into the catchment area. One simple strategy is route-based navigation where the routes consist of sets of nodes with overlapping catchment areas. We have investigated some methods for ensuring this overlap, but much more remains to be done [13, 36]. Beyond route-based navigation is complete topological navigation where route segments are concatenated together to form a graph [45]. There has been considerable work on this area, also known as topological SLAM in recent years [14, 15, 16, 17, 46]. We intend to apply the algorithm presented here both in route-based and topological navigation.

7. Conclusions

We have described a method for performing visual homing using the scale change of SIFT features. In fact, the method is not reliant on SIFT itself but requires features with an associated scale parameter. Numerous variants of the SIFT framework have been proposed and could be used for this purpose (e.g. SURF features [47]).

In this paper we have shown that homing in scale space performed significantly better than the warping method, which has been widely used as a benchmark in the field of visual homing. Future work will focus on demonstrations of the technique in 3D and improving robustness to field-of-view limitations.

8. Acknowledgements

Thanks to David Lowe for the use of his SIFT implementation and to Ralf Möller for his implementation of the warping method and for the use of image databases collected by his students.

9. References

[1] D. Churchill, A. Vardy, Homing in scale space, in: IEEE/RSJ International Conference on Robots and Systems (IROS), 2008, pp. 1307–1312.

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

[3] B. Cartwright, T. Collett, Landmark maps for honeybees, Biological Cybernetics 57 (1987) 85–93.

[4] A. Anderson, A model for landmark learning in the honey-bee, Journal of Comparative Physiology A 114 (1977) 335–355.

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

[6] P. Graham, V. Durier, T. Collett, The binding and recall of snapshot memories in wood ants (Formica rufa L.), Journal of Experimental Biology 207 (2003) 393–398.

[7] R. Morris, Spatial localization does not require the presence of local cues, Learning and Motivation 12 (1981) 239–260.

[8] S. Gillner, A. Weiss, H. Mallot, Visual homing in the absecne of feature-based landmark information, Cognition 109 (2008) 105–122.

[9] T. Collett, M. Collett, Memory use in insect visual navigation, Nature Reviews Neuroscience 3 (2002) 542–552.

[10] B. Kuipers, Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, Journal of Robotics and Autonomous Systems 8 (1991) 47–63.

[11] J. Hong, X. Tan, B. Pinette, R. Weiss, E. Riseman, Image-based homing, in: IEEE ICRA, 1991, pp. 620–625.

[12] A. Argyros, C. Bekris, S. Orphanoudakis, L. Kavraki, Robot homing by exploiting panoramic vision, Journal of Autonomous Robots 19 (1) (2005) 7–25.

[13] A. Vardy, Long-range visual homing, in: Proceedings of the IEEE International Conference on Robotics and Biomimetics, IEEE Xplore, 2006.

[14] M. Franz, B. Schölkopf, H. Mallot, H. Bülthoff, Learning view graphs for robot navigation, Autonomous Robots 5 (1998) 111–125.

[15] W. Hübner, H. Mallot, Metric embedding of view-graphs: A vision and odometry-based approach to cognitive mapping, Autonomous Robots 23 (2007) 183196.

[16] T. Goedemé, M. Nuttin, T. Tuytelaars, L. Van Gool, Omnidirectional vision based topological navigation, International Journal of Computer Vision 74 (3) (2007) 219–236.

[17] D. Filliat, Interactive learning of visual topological navigation, in: IEEE/RSJ International Conference on Robots and Systems (IROS), 2008.

[18] D. Dai, D. Lawton, Range-free qualitative navigation, in: IEEE ICRA, 1993.

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

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

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

[22] A. Vardy, R. Möller, Biologically plausible visual homing methods based on optical flow techniques, Connection Science 17 (1/2) (2005) 47–90.

[23] M. Zampoglou, M. Szenher, B. Webb, Adaptation of controllers for image-based homing, Adaptive Behavior 14 (2006) 245–252.

[24] R. Möller, Local visual homing by warping of two-dimensional images, Robotics and Autonomous Systems 57 (1) (2009) 87–101.

[25] R. Möller, M. Krzykawski, L. Gerstmayr, Three 2d-warping schemes for visual robot navigation, Autonomous Robots 29 (3) (2010) 253–291.

[26] A. Burke, A. Vardy, Visual compass methods for robot navigation, in: Proceedings of the Newfoundland Conference on Electrical and Computer Engineering, 2006.

[27] A. Vardy, A simple visual compass with learned pixel weights, in: Proceedings of the Canadian Conference on Electrical and Computer Engineering, IEEE Xplore, 2008.

[28] A. Rizzi, D. Duina, S. Inelli, R. Cassinis, Unsupervised matching of visual landmarks for robotic homing using Fourier-Mellin transform, in: Intelligent Autonomous Systems 6, 2000, pp. 455–462.

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

[30] K. Weber, S. Venkatesh, M. Srinivasan, Insect-inspired robotic homing, Adaptive Behavior 7 (1999) 65–97.

[31] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, R. Wehner, A mobile robot employing insect strategies for navigation, Robotics and Autonomous Systems, Special Issue: Biomimetic Robots 30 (2000) 39–64.

[32] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[33] S. Se, D. Lowe, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, The International Journal of Robotics Research 21 (8) (2001) 735–758.

[34] A. Briggs, Y. Li, D. Scharstein, M. Wilder, Robot navigation using 1d panoramic images, in: IEEE ICRA, 2006, pp. 2679–2685.

[35] J. S. Pons, W. Hübner, J. Dahmen, H. Mallot, Vision-based robot homing in dynamic environments, in: K. Schilling (Ed.), 13th IASTED International Conference on Robotics and Applications, 2007, pp. 293–298.

[36] A. Vardy, Using feature scale change for robot localization along a route, in: IEEE/RSJ International Conference on Robots and Systems (IROS), 2010.

[37] T. Röfer, Controlling a wheelchair with image-based homing, in: Proceedings of AISB Workshop on Spatial Reasoning in Mobile Robots and Animals, Manchester, UK, 1997.

[38] R. Möller, A. Vardy, S. Kreft, S. Ruwisch, Visual homing in environments with anisotropic landmark distribution, Autonomous Robots 23 (2007) 231–245.

[39] D. Churchill, Homing in scale space, Master's thesis, Memorial University of Newfoundland (2009).

[40] P. Royston, An extension of shapiro and wilk's w test for normality to large samples, in: Applied Statistics, 1982, pp. 115–124.

[41] P. Royston, Algorithm as 181: The w test for normality, in: Applied Statistics, 1982, pp. 176–180.

[42] J. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, Marcel Dekker Inc., New York, 1992.

[43] L. Kitchens, Basic Statistics and Data Analysis, Duxbury, 2003.

[44] E. L. Lehmann, Nonparametrics: Statistical Methods Based on Ranks, Holden and Day, San Francisco, 1975.

[45] M. Franz, H. Mallot, Biomimetic robot navigation, Robotics and Autonomous Systems, Special Issue: Biomimetic Robots 30 (2000) 133–153.

[46] S. Ferdaus, A. Vardy, G. Mann, R. Gosine, Comparing global measures of image similarity for use in topological localization of mobile robots, in: Proceedings of the Canadian Conference on Electrical and Computer Engineering, IEEE Xplore, 2008.

[47] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust features, Computer Vision and Image Understanding 110 (3) (2008) 346–359.