Robotica (2016) volume 34, pp. 2741–2758. © Cambridge University Press 2015
doi:10.1017/S026357471500034X

Homing with stereovision

Paramesh Nirmal and Damian M. Lyons*

Robotics and Computer Vision Lab, Fordham University, Bronx, NY, USA

(Accepted April 28, 2015. First published online: May 28, 2015)

SUMMARY

Visual Homing is a navigation method based on comparing a stored image of a goal location to the current image to determine how to navigate to the goal location. It is theorized that insects such as ants and bees employ visual homing techniques to return to their nest or hive, and inspired by this, several researchers have developed elegant robot visual homing algorithms. Depth information, from visual scale, or other modality such as laser ranging, can improve the quality of homing. While insects are not well equipped for stereovision, stereovision is an effective robot sensor. We describe the challenges involved in using stereovision derived depth in visual homing and our proposed solutions. Our algorithm, *Homing with Stereovision* (HSV), utilizes a stereo camera mounted on a pan-tilt unit to build composite wide-field stereo images and estimate distance and orientation from the robot to the goal location. HSV is evaluated in a set of 200 indoor trials using two Pioneer 3-AT robots showing it effectively leverages stereo depth information when compared to a depth from scale approach.

KEYWORDS: Visual homing; Visual navigation; Stereovision; Mobile robots; Computer vision.

1. Introduction

The designers of autonomous robots often find motivations and inspiration from biological systems. The study of insect navigation is subdivided according to different strategies and modalities that insects appear to employ. Two of the primary modalities are path integration (i.e., dead reckoning), and the use of visual landmarks. There is strong evidence from research on bees and ants that these insects use visual information to return to their nest or hive after a foraging trip.[15] Such insects are able to return to a previously visited place by comparing visual information gathered at their home location and the currently visible visual information. Thus, visual homing can be defined as the ability of an agent to return to a previously visited location (goal location) by comparing the image currently in the agent's view (current image) with a stored image of the goal location (goal image or *snapshot*).

Traditional robot navigation approaches focus on developing and maintaining an accurate geometric map of the environment.[5] Within this framework, moving to a goal location would be achieved by localizing both the goal and the current location within the map and planning a route between them. This approach requires localization and mapping, e.g., SLAM.[20] Navigation approaches based on SLAM are concerned with accurately assessing the location of the robot, the goal, and other entities in the environment and planning an accurate, collision-free path. The quality and robustness of the path depends heavily on the accuracy of the map and on the localization of the robot on the map. Visual homing does not require a map, or the localization of the robot to a map. It thus avoids both the computational load of mapping and localization, and also the path errors that might result from any inaccuracies in those processes. However, visual homing is restricted to cases where the desired goal (home) location is in view. It cannot be used in long-distance navigation in general therefore or in any navigation where the goal location is completely occluded. Nonetheless, visual homing can be an effective and robust approach for short-distance navigation.

In this paper, we present a novel approach to visual homing that combines SIFT-based feature matching and stereo distance measurements to achieve fast and accurate visual homing. Insects do

* Corresponding author. E-mail: dlyons@fordham.edu

not have the anatomical infrastructure for effective stereovision, having relatively closely spaced, immobile eyes with fixed focus-optics.[18] However, they employ other approaches to gauge depth and it seems very reasonable to include depth in the information available for visual homing (as do[3] from visual scale and[2] from laser data). We describe the challenges involved in effective use of depth from stereovision to improve homing performance and our proposed solutions. We evaluate our approach using a set of 200 experimental runs using two pioneer 3-AT mobile robots, and we compare the performance of our approach with that of another recent visual homing algorithm which incorporates depth information from visual scale information. We show the stereovision approach achieves faster and more accurate performance for this set of experimental runs but at the cost of constructing a wide field of view stereo image.

The paper is structured as follows. The next section reviews related work on the visual homing problem to place this research in context. Section 3 then presents our approach, the HSV algorithm. Section 4 describes the experimental methods and an experimental evaluation of this approach is presented in Section 5. Section 6 presents a discussion of our results and we conclude in Section 7 with a comparison of our method with other visual homing methods as well as discussion of future work.

2. Related Work

Existing visual homing methods can be divided into two main categories: holistic and correspondence based, and we review each category below.

2.1. Holistic approach

The holistic approach treats the image as a whole, performing the comparison of the current and goal images using one of three whole-image techniques: warping methods, parameter methods, and DID (descent in image distance) methods.

The *warping method*, originally proposed by Franz *et al*.,[8] warps the current image according to certain movement vectors. The search space is the space of movement. The goal movement vector is the movement vector that produces a warped image that is most similar to the goal image. This approach operates under the assumptions that all objects in the environment are approximately equidistantly far with respect to the robot, and that the scene is "planar", so the depth different between objects in the environment is small compared to the distance from the robot to the objects. Such assumptions are rarely completely fulfilled in practice;[4] however, in environments where all the landmarks are relatively distant, the assumptions are reasonable. Franz's warping method was proposed for 1D images. However, Möller[13] extended Franz's approach for processing 2D images.

Parameter methods operate on the assumption that the agent does not have a stored image of the goal location. Instead, the agent has stored an epitomized description of the landmarks in the goal image.[12] It has been hypothesized since the early 1980s (Wehner)[23] that biological agents such as desert ants and honeybees store images in the form of condensed parameter descriptors. Parameter methods use very little memory and processing time because they do not process full images. Their simplicity is an additional advantage as a model of insect visual homing.

Descent in image distance methods rely on the fact that the distance measure between two images taken at different locations increases smoothly with increasing spatial distance (Zeil *et al*.)[24] Möller and Vardy[14] leverage this fact for local visual homing using matched-filter descent in image distance. Estimating the spatial gradient using two matched-filters on the image planes produces the *home vector*, which is used to lead the agent back to the goal location. This method operates under the assumption that there is no change in orientation from the agent's goal location and current location.

2.2. Feature-based approach

The feature-based approach, also known as the *correspondence approach*, establishes correspondences between regions or features in the current image and those in the goal image. This approach is based on feature detection algorithms and algorithms that match the features from the current image to the goal image. Each pair of correspondences describes a *shift vector*,[3] representing the shift of features in 2D image space from one image to the other. This vector can be transformed into a *home vector*, which can be used to navigate the agent back to its home location.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

Much work has been done to establish correspondences between two images for homing purposes. Vardy and Möller[21] apply optical flow techniques for visual homing. Block matching, and differential techniques have been tested and found to be robust, accurate homing methods, despite their simplicity. Cartwright and Collet[1] use high contrast features in landmarks to determine correspondences between the two images. Other feature-based methods utilize Harris corners.[22]

In the past few years, the use of SIFT (Scale Invariant Feature Transform) to determine correspondence between the current and goal image has gained great popularity due to the robustness of their descriptor vectors with respect to changes in scale, rotation, and lighting.[11] Pons *et al.*[17] use SIFT features to detect visual landmarks in the scene, followed by a matching and voting scheme of the detected features. The homing algorithm presented in[17] is driven under the assumption, as were Vardy and Möller,[21] that the current and goal images both have the same orientation.

Recently, visual homing methods have been extended by fusing additional data with the visual image. Sturzl and Mallot[19] developed a low-resolution panoramic stereo camera (72 disparities) and demonstrated that even this amount of extra information could improve homing.

Homing in Scale Space (HiSS) is a feature-based visual homing method developed by Churchill & Vardy[3] that compares the image size of visual features in current and goal images, an estimate of depth, to determine whether the robot has to move towards or away from the features to approach the goal. The algorithm is based on extracting and matching SIFT features from the goal image and the current image. Each SIFT feature, $\boldsymbol{f}$, is described by:

$$\boldsymbol{f} = \left\{ f^u, f^v, f^\sigma, f^\rho, \boldsymbol{f^d} \right\}, \tag{1}$$

where f^u, f^v are the feature's location in the image in **2D** image-based coordinates, f^σ is the feature's scale, f^ρ is the orientation of the feature and $\boldsymbol{f}^d$ is the feature's keypoint descriptor, which is a 128-dimensional vector used for feature matching.

In HiSS, the robot is equipped with an omnidirectional camera, thereby ensuring that the goal location can be seen for any robot orientation. The horizontal position coordinate of the features in the omnidirectional image is used to determine the angular change in heading. To determine distance to the goal, the scale component of the SIFT keypoint vector is leveraged. HiSS determines whether the robot is closer to or further from the goal location by checking whether the features in the image have expanded or contracted as measured by the feature's change in scale. The scale component f^σ is discrete[11] with a small range of values—a rough depth measurement. To home despite the lack of accurate knowledge about distance to the goal, the robot takes small steps towards or away from features in the scene until the current image and the goal image satisfy a certain degree of similarity.[4] The algorithm operates on the assumption that the features in the scene are distributed evenly on either side of the image, thus half of the features expand while the other half contract.[3]

Choi *et al.*[2] implement homing by fusing range information from a laser rangefinder with the visual image information. They demonstrate that using this additional sensory information enables autonomous homing of a robot in large-scale indoor environments, as opposed to small-scale local environments. Jin and Xie[9] use a stereo camera to implement robot hand-eye coordination, navigating the humanoid robot's arm from a displaced location back to its original location, but assumes a fiducial marker is present on the robot's arm.

This prior work shows that additional information on feature depth improves homing. The advantages of using stereo camera depth rather than laser ranging include the simplicity of a single sensor, and the relative ease of registration of depth and image data. The advantage over visual scale includes the accuracy and resolution of depth. However, the field of view of a stereo camera system is typically small, limited to just the overlap of the fields of view of each camera in the stereo pair, whereas visual homing is typically done with a panoramic camera to ensure the home location remains visible. A key challenge therefore is to build a stereo camera approach that has the necessary field of view while maintaining its advantages of resolution and accuracy.

It may also appear that depth information renders the visual homing problem trivial if it can directly extract the distance and direction towards the home location! However, the depth information available is only that to observed features, and the distance to the goal can still only be inferred; even worse as we will show, a direct vector to the goal cannot be calculated for every case, raising the challenge of developing a convergence strategy. In,[16] we proposed using a stereo camera for homing,

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

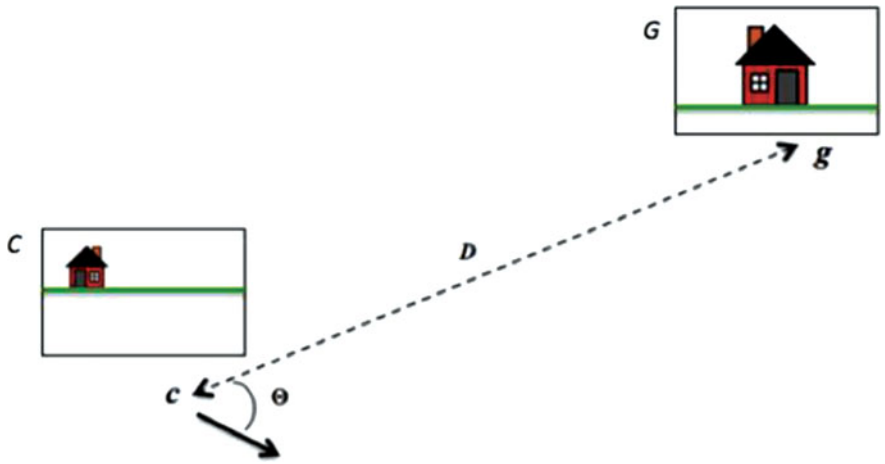


Fig. 1. Robot's current location marked by c, robot's goal location marked by g. The angle Θ is the amount the robot has to rotate and D is the distance the robot has to travel to reach g from c.

where the goal image was constructed by panning the stereo camera. In this paper, we build on that idea, presenting a detailed algorithm for homing using stereovision, a proof of homing convergence, and a performance evaluation based on 200 trials using two Pioneer 3-AT robots.

3. Homing with Stereovision

The proposed method presented in this paper—- HSV, is a feature based method. It extends existing visual homing methods by including depth information from a movable stereo camera mounted on the robot for distance estimation in addition to using the visual information from the camera for feature matching. HSV relies upon SIFT to find correspondences between features but differs from HiSS in that instead of using the scale component to determine whether features have moved closer or further from the robot, HSV utilizes information from the stereo camera to obtain a more accurate measurement.

We begin in Section 3.1 by introducing the notation used to describe the algorithm. Section 3.2 will state the assumptions on which the algorithm is based. The algorithm is described in Sections 3.3 and 3.4. Finally, a convergence guarantee is presented in Section 3.5.

3.1. Notation

Let C be the image in the robot's current view and G be the image of the goal location (also known as the *snapshot*). Let $\boldsymbol{c}$ be the robot's current position, and $\boldsymbol{g}$ be the robot's position at the goal location. The home vector:

$$\boldsymbol{h} = [\Theta, D], \tag{2}$$

is the vector whose components are the angle Θ the robot has to rotate and the distance D the robot has to move from $\boldsymbol{c}$ to $\boldsymbol{g}$.

Figure 1 shows a top view of $\boldsymbol{c}$ and $\boldsymbol{g}$, and the home vector components Θ and D. By first rotating Θ degrees, which is the change in orientation between the robot's current and goal position, and then by travelling distance D, which is the straight-line distance between $\boldsymbol{c}$ and $\boldsymbol{g}$, the robot can home in a single step.

HSV estimates Θ and D; However, it is not possible in general for the robot to reach its goal in one step using the estimated values. Instead, as we will explain (Section 3.3), the robot will only move by a fraction of Θ and D in each step.

3.2. Assumptions

An important though seemingly obvious assumption on which the HSV algorithm is based is that the goal location must be in the robot's view. If the robot cannot see the goal location, then visual

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

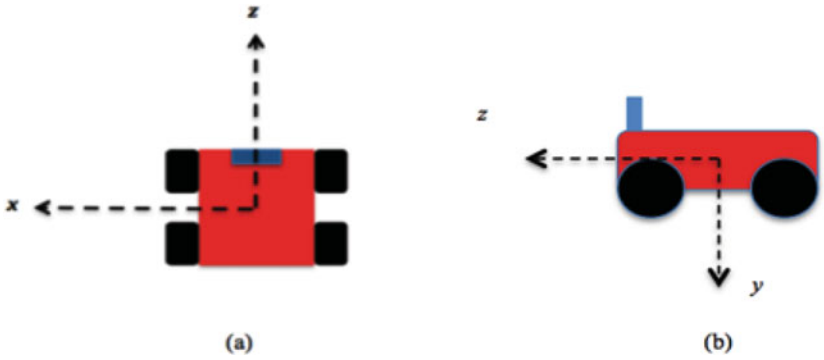


Fig. 2. (a) x and z coordinate frame with respect to the robot (b) y and z coordinate frame with respect to the robot.

homing cannot take place. This assumption posits an upper bound on the distance between $\boldsymbol{c}$ and g for the HSV algorithm to commence. The upper bound is the range of the robot's visual sensor. If the robot's visual sensor has a large range (as is the case with a stereo camera), then the distance between $\boldsymbol{c}$ and g is bounded by the distance to the visual horizon, since the robot cannot see g if g is past the visual horizon.

The HSV algorithm will assume that there are no obstacles between the robot's current position and the goal position. Since the algorithm is based on wide-field image comparison, in fact a small amount of obstacles and occlusion will not significantly affect the calculation of the home vector. It is also assumed that the robot's current position $\boldsymbol{c}$ and goal position g lie on the same horizontal plane.

The HSV algorithm is dependent on a visual characteristic of the environment namely, the number of SIFT features in the environment. The environment in which the robot operates must be abundant in features to obtain enough SIFT matches. The best case is when the environment is saturated with features; this will be discussed in the following sections. Since visual homing is a method to return to a previously visited location, it is fair to assume that the agent has the goal image G stored in memory.

3.3. Algorithm

The principal steps in the HSV algorithm are:

- Determine the home vector $h = [\Theta, D]$ from a comparison of the current and goal images.
- Rotate the robot by Θ, and translate the robot by D.

3.3.1. Determining the home vector. HSV leverages information from a stereo camera image to calculate $\boldsymbol{h}$. In overview, a set of SIFT features is calculated from the stereo camera's current left visual image, and the SIFT features are augmented with distance information from the stereo camera's depth image. The home vector will be calculated by matching the features to SIFT features previously extracted from the goal image.

Each SIFT feature $\boldsymbol{f}$ can be defined as:

$$\boldsymbol{f} = \left\{ f^x, f^y, f^z, f^\alpha, \boldsymbol{f}^d \right\}, \tag{3}$$

where (f^x, f^y) is the feature location in *mm* in the real world with respect to the robot, with the robot being the centre of the coordinate system. f^z is the feature's *depth* in *mm*, i.e., the distance from the centre of the robot to the feature in the z-plane. And finally, f^α is the angular position (azimuth) of the feature f with respect to the current orientation of the robot. The vector $\boldsymbol{f}^d$ remains as the feature descriptor. Figure 2 shows the coordinate frame used in the algorithm, and Fig. 3 shows the location of a feature $\boldsymbol{f}$ and it's coordinates with respect to the robot's coordinate frame. It is possible for the position or depth of the feature, (f^x, f^y, f^z), obtained from the stereo camera to be an inaccurate reading due to stereo related errors for individual pixels. To overcome this difficulty, a set of points around the feature is collected, and the RANSAC[7] algorithm is performed on that set of points to obtain consistent readings for (f^x, f^y, f^z).

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

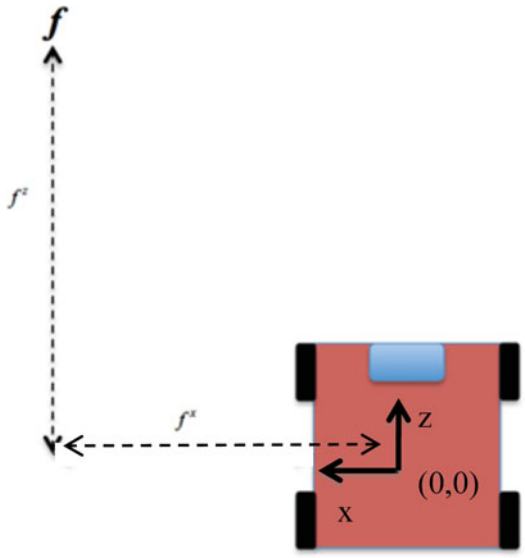


Fig. 3. Location of scene element giving rise to feature f with respect to the robot. The features x and z position are denoted by f^x and f^z, respectively.

Note that, HSV only makes use of information delivered by the stereo camera; the algorithm does not make use of the robot's odometer, or any other sensor modules to determine robot position.

For each matched feature (i, j), where the ith feature in G is matched with the jth feature in C, the lengths $G^{i,\alpha}$ and $C^{j,\alpha}$ (also in *mm*) are calculated as:

$$G^{i,\alpha} = \arctan\left(\frac{G^{i,x}}{G^{i,z}}\right), \tag{4}$$

and

$$C^{j,\alpha} = \arctan\left(\frac{C^{j,x}}{C^{j,z}}\right), \tag{5}$$

where $G^{i,x}$ is the x-coordinate of the location of feature i of image G, and $G^{i,z}$ is the depth of feature i of image G. Similarly, $C^{j,x}$ is the x-coordinate of the location of feature j of image C, and $C^{j,z}$ is the depth of feature j of image C.

If M is the set of matched features between C and G, and $m = (i, j) \in M$, where M is the set of all matched features m, and where $\theta^m = ccw(G^{i,\alpha} - C^{j,\alpha})$, is the counter-clockwise angle of difference (since features can "wrap-around" the 270° field of view), then we calculate the average angular difference between the two images as:

$$\Theta = \frac{1}{|M|}\sum_{m\in M}\left(\theta^m\right), \tag{6}$$

θ^m is the difference in angular position for a single matched feature $m = (i, j)$. Θ, the average of all θ^m, is the angular change in orientation through which the robot will rotate to face $\boldsymbol{g}$.

Similarly, the distance D the robot has to move is calculated as the average of the difference in depth for all matched features, where $d^m = G^{i,z} - C^{j,z}$ is the difference in depth for a single matched feature $m = (i, j) \in M$:

$$D = \frac{1}{|M|}\sum_{m\in M}\left(d^m\right). \tag{7}$$

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

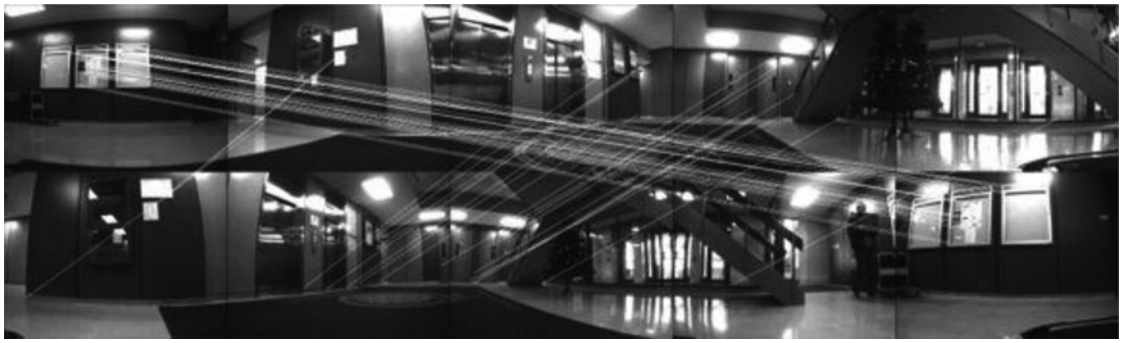


Fig. 4. A wide-field goal image (top) with SIFT features matched against a wide-field current image (bottom) using Lowe's SIFT tools. Matched features are connected with a straight line. Images are taken over a 270° pan.

Because of the need in visual homing to be able to see the goal location, it is common to use an omnidirectional camera, maximizing the field of view. Stereo cameras in contrast have a relatively small field of view, replying as they do on the overlap in fields of view of their component cameras.

To address this challenge, we construct G and C as composite wide-field images, i.e., images composed of the several images taken by panning the stereo camera over a wide range, and concatenating the collected images. Figure 4 shows an example of a wide-field goal image matched with a wide-field current image.

The wide-field images taken by the robot are not perfect panoramas in that the component images in the concatenated image exhibit some overlap. It is possible to perform an image-stitching algorithm on the set of images taken by the stereo camera to produce a panoramic image and a panoramic stereo image. Investigating this, we conclude that stitching is not necessary and in fact not doing it can improve performance:

(1) Panoramic stitching can be computationally expensive therefore increasing time taken to process each image.
(2) By allowing overlapping scenes, some features exist twice in the composite image thus increasing the likelihood of the feature being matched with a feature in another image.

The composite wide-field images encompass a 270° stereo field of view. This restriction is based on the maximum range of the pan-tilt unit on which the stereo camera is mounted. It is possible to capture a wide field image that encompasses a smaller field of view (<270°) however, we choose to use the maximum possible field of view to maximize the number of SIFT features being detected and matched.

3.3.2. Moving to the home location. Once $\boldsymbol{h}$ has been calculated, the robot could immediately rotate Θ degrees and translate D mm. However, if this single move approach is implemented, the resulting homing motion may be inaccurate, as we explain in detail in the next section. Instead, the robot only moves by a fraction of Θ and D after each image comparison, and from this intermediate location it takes another image and iterates the calculations. The robot can take several steps to reach $\boldsymbol{g}$. At each step, the robot will move by a fraction of Θ and D until a termination condition is satisfied indicating it has arrived at $\boldsymbol{g}$. Angular and translation gains to determine this fraction, g_a and g_d respectively, were empirically determined, and these are discussed in the next section. The robot cannot use odometry information to determine whether $\boldsymbol{c} = \boldsymbol{g}$. Therefore, a termination condition is implemented that determines whether the robot has reached $\boldsymbol{g}$ based upon the value of Θ and D. The termination condition, $\boldsymbol{T}$ is:

$$(|\Theta| < \varepsilon) \wedge (|D| < D_{\text{tol}}), \tag{8}$$

where ε is a small angle and D_{tol} is a small distance tolerance. A convergence guarantee for this approach is presented in Section 3.5. The HSV algorithm is summarized below.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

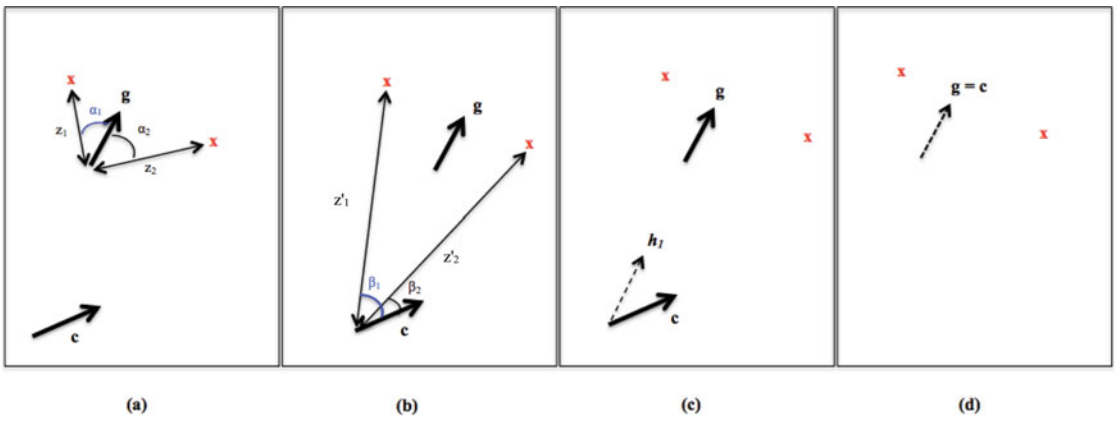


Fig. 5. Detected features where a single movement vector is the single step[1] *home vector*. Panel (a) shows the goal position, g with respect to features marked as "x". Panel (b) shows the current position, c with respect to the features. In panel (c) robot would rotate by Θ_1 (dotted line). In panel (d) robot position after translating D_1 and goal is reached since $\boldsymbol{c} = g$.

HSV$(G,\ C,\ \varepsilon,\ D_{\text{tol}})$:

1. **Do:**
2. Capture current image C;
3. M = SIFT_match$(G,\ C)$;
4. $\Theta = 0;\ D = 0;$
5. **For** each $\boldsymbol{m} \in \boldsymbol{M}$ where $\boldsymbol{m} = (i, j)$:
   i. $\Theta += C^{j,\alpha} - G^{i,\alpha}$;
   ii. $D += C^{j,z} - G^{i,z}$;
6. $\Theta = {}^{\Theta}/_{|M|};\ D = {}^{D}/_{|M|};$
7. Rotate robot by $g_a\Theta$ and translate by $g_d D$;
8. **While** $(|\Theta| > \varepsilon$ **AND** $|D| > D_{\text{tol}})$;

Summary of Parameters used in the Algorithm

g_a *angular gain,* $1/3$

g_d, *translational gain,* $1/2$

ε, *angular termination tolerance,* 5°

D_{tol}, *distance termination tolerance,* 30 cm

3.4. Homing in a single step

It may not be possible to move to the home location accurately in one move because of the relationship between the homing angle and distance, and we present two examples below to explain this restriction. Consider the detected features in Fig. 5. The robot would exhibit rotation $\Theta_1 = \frac{1}{2}(\theta_1 + \theta_2)$ and movement $D_1 = \frac{1}{2}(\text{dif}(z_1) + \text{dif}(z_2))$,[2] where $\theta_1 = \alpha_1 - \beta_1$, and $\theta_2 = \alpha_2 - \beta_2$. Let $\boldsymbol{h}_1 = [\Theta_1, D_1]$. The vector $\boldsymbol{h}_1$ would be a suitable *home vector*, as it could precisely lead the robot back to the goal location (assuming no error in motion).

Now, consider the detected features in Fig. 6. The robot would rotate by $\Theta_2 = \frac{1}{2}(\theta_1 + \theta_2)$ and movement $D_2 = \frac{1}{2}(\text{dif}(z_1) + \text{dif}(z_2))$, where $\theta_1 = \alpha_1 - \beta_1$, and $\theta_2 = \alpha_2 - \beta_2$. Let $\boldsymbol{h}_2 = [\Theta_2, D_2]$. It is evident that in this case, rotating by Θ_2 and then translating the robot by D_2 will *not* accurately lead the robot back to its goal location. However, if the robot moves by a fraction of $\boldsymbol{h}_2$ and the procedure is iterated, then the algorithm will converge to $\boldsymbol{g}$, resulting in accurate homing.

[1] The *single step home vector* is the home vector that leads the robot back to its home location in a single step.
[2] $\text{dif}(z_1) = G^z - C^z$, where G^z and C^z are the depths of the matched feature in the goal image and the current image, respectively.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

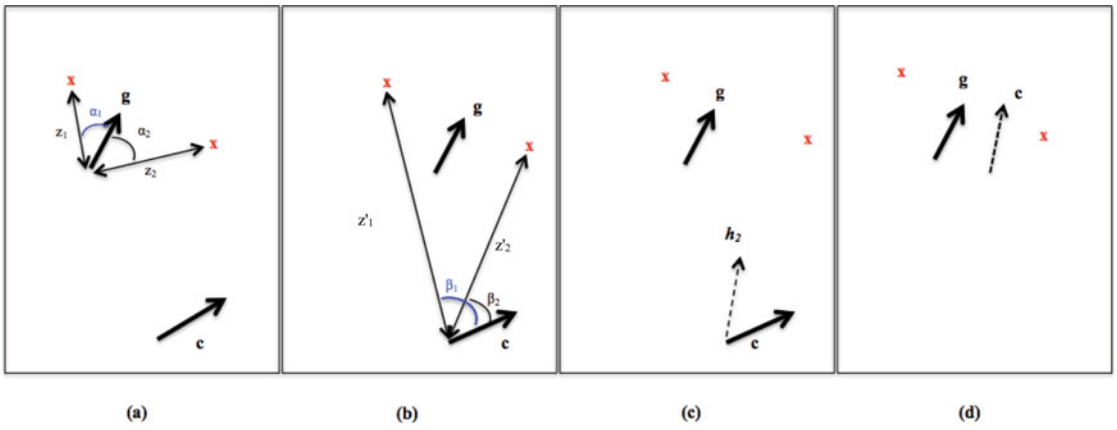


Fig. 6. Detected features where a single movement vector is not the single step *home vector*. Panel (a) goal position $\boldsymbol{g}$ with respect to features marked as "x". Panel (b): current position $\boldsymbol{c}$ with respect to features. Panel (c): robot exhibiting rotation Θ_2. Panel (d): robot position after translation motion D_2, goal is not reached because $\boldsymbol{c} \neq \boldsymbol{g}$.

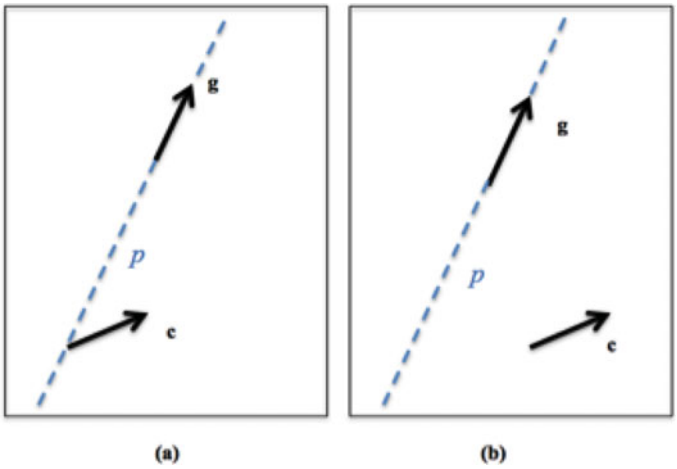


Fig. 7. Panel (a): single step case where the line p (back tracing position $\boldsymbol{g}$) intersects $\boldsymbol{c}$, in the single step case the robot can accurately reach its home location in one step. Panel (b): common case where line p does not intersect $\boldsymbol{c}$, In this case the robot cannot accurately home in one step.

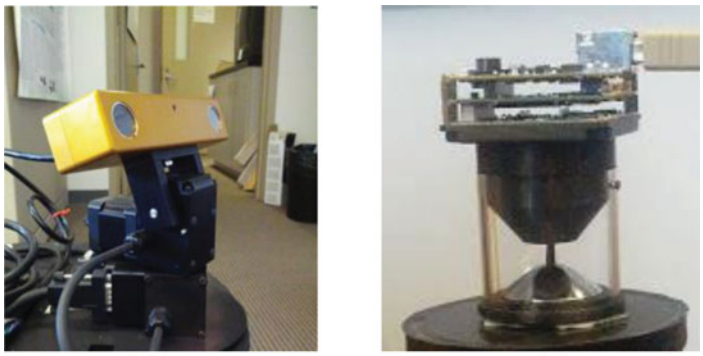
Figure 7(a) summarizes the situation in which a single move *is* possible (as in the example in Fig. 5). Figure 7(b) by contrast summarizes the situation in which a single move is *not* possible (as in the example in Fig. 6).

To address this problem, Θ and D are scaled by an angular gain and a distance gain, respectively, before moving the robot. The robot rotates by $g_a \cdot \Theta$ and translates by $g_d \cdot D$, where $g_a \in (0, 1]$ is the angular gain, and $g_d \in (0, 1]$ is the distance gain. A final robot position within 30 cm of the goal $\boldsymbol{g}$ can be achieved by choosing $g_a = 1/3$, along with $g_d = 1/2$ for the indoor experiments reported in this paper. In the next section, we present a convergence guarantee for this incremental motion strategy.

3.5. Convergence

Theorem. Let, $\bar{\Theta} = \Theta_1, \Theta_2, \Theta_3..\Theta_N$ be the sequence of angles returned by the HSV algorithm, where N is the number of iterations for which the algorithm runs. Let $\bar{D} = D_1, D_2, D_3..D_N$ be the sequence of distances returned by the HSV algorithm for the N iterations. If all the assumptions in Section 3.2 hold, if the SIFT feature matching is error free, if all depth values are accurate, and if the robot achieves the commanded displacements at each step, then HSV is guaranteed to terminate after N steps with $(\Theta_N < \varepsilon) \wedge (D_N < D_{\text{tol}})$.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press



(a) (b)

Fig. 8. (a) BumbleBee2 Stereo Camera mounted on Directed Perception pan-tilt (b) IT + R omnidirectional camera.

Proof. Consider Θ_k and Θ_{k+1} in $\bar{\Theta}$. Let α_k be the actual orientation of the robot at step k of the sequence, in which case we can say that HSV sets $\alpha_k = \alpha_{k-1} - \Theta_{k-1}$. Let α_g be the robot orientation when it is at the goal (home) location. At each step *k,* HSV calculates the home vector $[\Theta, \mathrm{D}]$ according to Eqs. (6) and (7). Each Θ_k is calculated as $\Theta_k = \mathrm{g}_a\ \Theta$. Given our assumptions about the visibility of the goal, the correctness of feature matching and of depth, we can say this is an accurate estimate of the rotation to goal, $\Theta = \alpha_k - -\alpha_g$, and so $\Theta_k = g_a(\propto_k - \propto_g)$.

Under these conditions, each Θ_k is smaller than or equal to the one before it:

$$
\begin{aligned}
&\Theta_{k+1} \leq \Theta_k \\
&\quad \Rightarrow \left(\propto_{k+1} - \propto_g\right) \leq (\propto_k - \propto_g), \quad \text{substituting for } \Theta_k \text{ and dividing out } g_a. \\
&\quad \Rightarrow \left((\alpha_k - g_a(\propto_k - \propto_g)) - \propto_g\right) \leq (\propto_k - \propto_g), \text{ substituting for } \propto_{k+1} \text{ and then for } \Theta_k, \\
&\quad \Rightarrow (1 - g_a)(\propto_k - \propto_g) \leq (\propto_k - \propto_g), \quad \text{by rearrangement.}
\end{aligned}
$$

In the case that $\propto_k = \propto_g$ then the two sides are equal; in all other cases, Θ_{k+1} is strictly less than Θ_k, since $g_a \varepsilon\, [0, 1]$. The HSV algorithm terminates when $(\Theta < \varepsilon) \wedge (D < D_{\mathrm{tol}})$. Since we have shown that $\Theta_{k+1} < \Theta_k$ and since in the case that $\propto_k = \propto_g$, $\Theta_k = 0$, then 0 is the greatest lower bound for the sequence. In that case, we must eventually have $\Theta_k < \varepsilon$ for some k and therefore ε will be the greatest lower bound for the sequence when the termination condition is enforced.

By a similar approach we can show that $D_{k+1} \leq D_k$ for $g_d \varepsilon\, [0, 1]$, and that eventually we must have $D_k < D_{\mathrm{tol}}$ for some k. Hence eventually we must have $(\Theta < \varepsilon) \wedge (D < D_{\mathrm{tol}})$, guaranteeing convergence.

4. Experimental Setup

This section describes the experimental methodology followed in evaluating the performance of the HSV algorithm and the HiSS algorithm with the objective of comparing the two.

4.1. Experimental trials

The experimental trials were conducted using a Pioneer 3-AT robot (P3). The P3 was equipped with a Point Grey BumbleBee2 stereo camera mounted on a Directed Perception Pan-Tilt Unit (shown in Fig. 8(a)). The HiSS algorithm was tested on a P3 equipped with an IT + R omnidirectional camera (shown in Fig. 8(b)). The omnidirectional camera is comprised of a camera mounted upwards facing a hyperbolic mirror. Figures 9 and 10 show a map and a pictorial description of the testing area, an indoor area with a smooth, flat floor.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

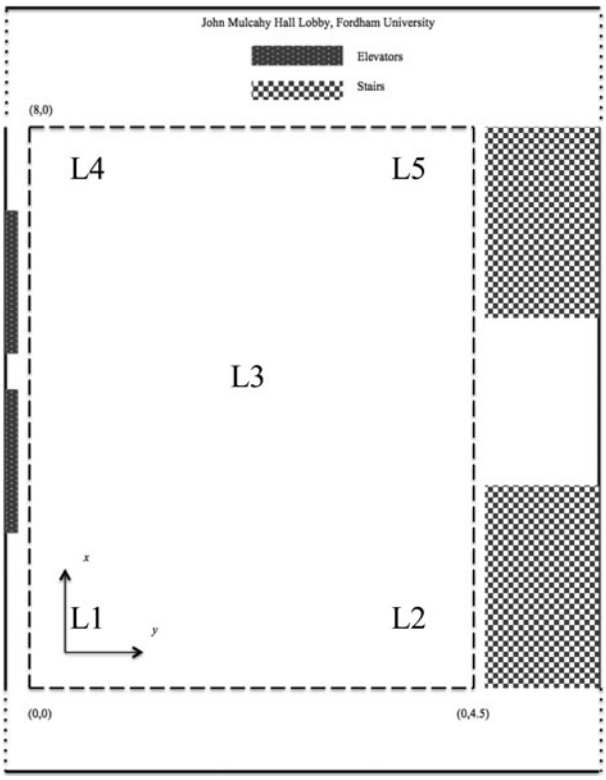


Fig. 9. Map of testing area. Testing region shown enclosed in dotted lines with test locations L1 through L5. Units are in meters.

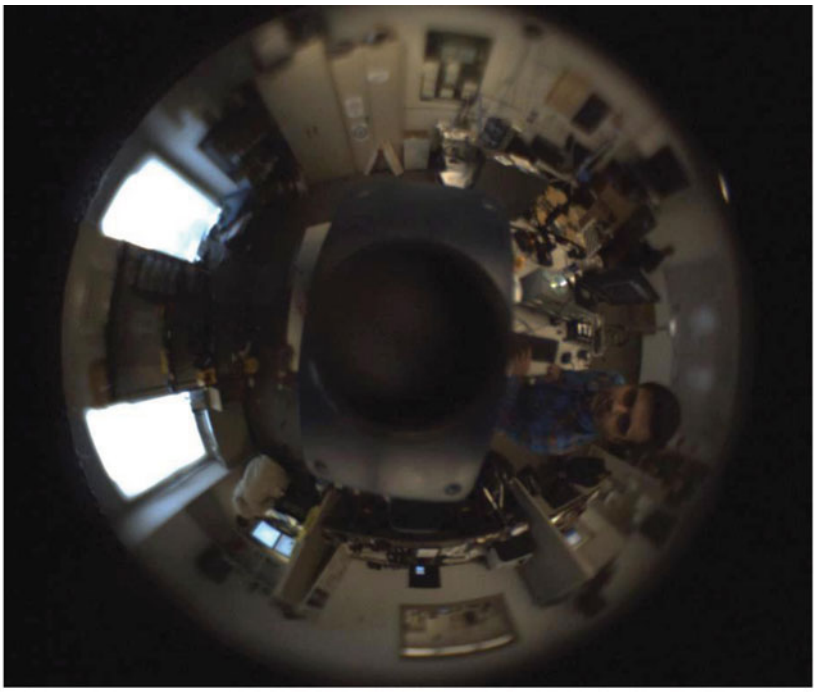
Fig. 10. Testing area (Lobby of JMH Building, Fordham University).

4.2. Configuration

Both the HSV and HiSS algorithms were implemented in *C++* using the OpenCV[3] library. The Pioneer 3-AT robots have an on-board computer with an Intel Dual-Core processor; both algorithms were tested on these processors. The commands to rotate and move the robot were implemented

[3] Obtained from http://opencv.org/

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press



(a)

(b)

Fig. 11. (a) Warped image returned from omnidirectional camera (b) resultant image from unwarping process.

using the Aria *C++* SDK supplied with the Pioneer 3-AT. Both algorithms calculated angular and translational displacements and used the same Aria commands and velocity profiles to move the robots the commanded amounts.

For the HSV algorithm, the stereo camera took five images at pan increments of 67.5°. So for each location the robot visited, it captured images with its stereo camera panned at $\{-135°, -67.5°, 0°, +67.5°, +135°\}$. The five images were concatenated to produce a composite wide-field image, which encompassed a 270° field of view around the robot. Moreover, the stereo camera was tilted upwards at 5° to allow more features from the walls to be present in the image.

The image captured by the omnidirectional camera (for the HiSS algorithm) was a warped image, and required unwarping using the imaging software supplied with the IT + R camera for the HiSS algorithm to be able to process the image. Figure 11 shows a warped image and the resultant image from the unwarping process. The image used by the HiSS algorithm has dimensions 1476 × 225 pixels. To reduce address caused by experimental bias, both algorithms worked with the same picture resolution and thus both algorithms work on roughly the same number of SIFT features. The composite wide field image used by the HSV algorithm was resized to a width of 1476 pixels. The SIFT tools for both algorithms were configured with the ratio of scores from best to second best feature match being increased from 0.6 to 0.8. Lowe[11] states that this change results in a small decrease in match accuracy while dramatically increasing the number of matches.

The HSV algorithm is configured with angular gain $g_a = 1/3$ and distance gain $g_d = 1/2$ to keep the position error at a minimum when the robot reaches $\boldsymbol{g}$. The HiSS algorithm was setup to run a

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

constant distance of 400 mm at each step. This reduces the number of steps the HiSS algorithm takes to reach $\boldsymbol{g}$ whilst preventing overshooting the goal position. Both algorithms were allowed to run for (at most) 14 steps, and the position of the robot recorded at each step. The trials were terminated on the 14th step, or in the case of collisions with a wall or other obstacles.

5. Results

This section presents the results of the experimental trials described in the previous section. It also describes a set of performance measures used to compare the two algorithms.

5.1. Performance measures

Four different performance metrics are defined to measure the accuracy of the visual homing algorithms.

5.1.1. Position error. The visual position error, ε_p is the Euclidean distance from the centre of the robot (at its final location) to the goal location $\boldsymbol{g}$.[16] Since HiSS[3] does not include a termination condition, the position of the robot is recorded during each step in the algorithm, and the closest position to the goal location is used as the goal position returned by the HiSS algorithm. The algorithm runs for 14 steps as stated in Section 3.2. The position error chosen for the HSV algorithm is defined as:

$$\varepsilon_p = \min\left(\varepsilon_p^{\text{det}}, \varepsilon_p^{\text{undet}}\right), \tag{9}$$

where $\varepsilon_p^{\text{det}}$ is the position error of a detected return and $\varepsilon_p^{\text{undet}}$ is the position error of an undetected return (after the maximum 14 steps). A detected return occurs if the algorithm terminates when reaching the goal location whilst an undetected return occurs ifs the algorithm continues to run even though the robot has reached the goal location.

5.1.2. Angular error. The visual homing angular error, ε_θ is defined as:

$$\varepsilon_\theta = \left|\Theta_{\text{ideal}} - \Theta_{\text{homing}}\right|, \tag{10}$$

where Θ_{homing} is the angle computed by the homing algorithm in the first iteration and Θ_{ideal} is the single rotation the robot must exhibit in order to reach $\boldsymbol{g}$ from $\boldsymbol{c}$. Θ_{ideal} is manually set as the change in heading between $\boldsymbol{c}_0$ and $\boldsymbol{g}$. This step is taken to make sure our results cover a range of homing angles in the set:

$$\mathcal{A} = \{30, 45, 60, 90, 120\}. \tag{11}$$

5.1.3. Return ratio. The return ratio *RR* is the ratio of successful returns to home location to unsuccessful. A successful return occurs when the robot returns to within 30 cm of $\boldsymbol{g}$.

5.1.4. Number of steps. The number of steps N is the number of iterations of the algorithm taken by the robot to reach $\boldsymbol{g}$ from the start position $\boldsymbol{c}_0$. Since the robot is only allowed to run for (at most) 14 steps for each algorithm, $N \leq 14$. Since both algorithms use the same Aria velocity profile to move, the only differences lies in the size and number of steps chosen.

5.2. Results

A set of five goal positions and five start positions were chosen distributed across the testing region as shown in Fig. 9. For each position, five trials were conducted and the average for each performance measure listed in Section 5.1 was computed. The set of positions, $\wp$ used for the experiments is defined below (in metres):

$$\wp = \{L1 = (1.0, 1.0), L2 = (1.0, 3.5), L3 = (4.00, 2.25), L4 = (7.0, 1.0), L5 = (7.0, 3.5)\}. \tag{12}$$

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

Table I. Index labels for each trial of goal position and start position.

Index	***(gx, gy),(cx, cy)***
1	L1 = (1.0,1.0), L2 = (1.0,3.5)
2	(1.0,4.0),(1.0,2.25)
3	(1.0,7.0),(1.0,1.0)
4	(1.0,7.0),(1.0,3.5)
5	(1.0,4.5),(3.5,2.25)
6	(1.0,1.0),(3.5,1.0)
7	(1.0,7.0),(3.5,1.0)
8	(1.0,7.0),(3.5,3.5)
9	(7.0,4.0),(1.0,2.25)
10	(7.0,1.0),(1.0,1.0)
11	(7.0,1.0),(1.0,3.5)
12	(7.0,7.0),(1.0,3.5)
13	(4.0,1.0),(2.25,1.0)
14	(4.0,7.0),(2.25,1.0)
15	(4.0,1.0),(2.25,3.5)
16	(4.0,7.0),(2.25,3.5)
17	(7.0,1.0),(3.5,1.0)
18	(7.0,7.0),(3.5,1.0)
19	(7.0,4.0),(3.5,2.25)
20	(7.0,1.0),(3.5,3.5)

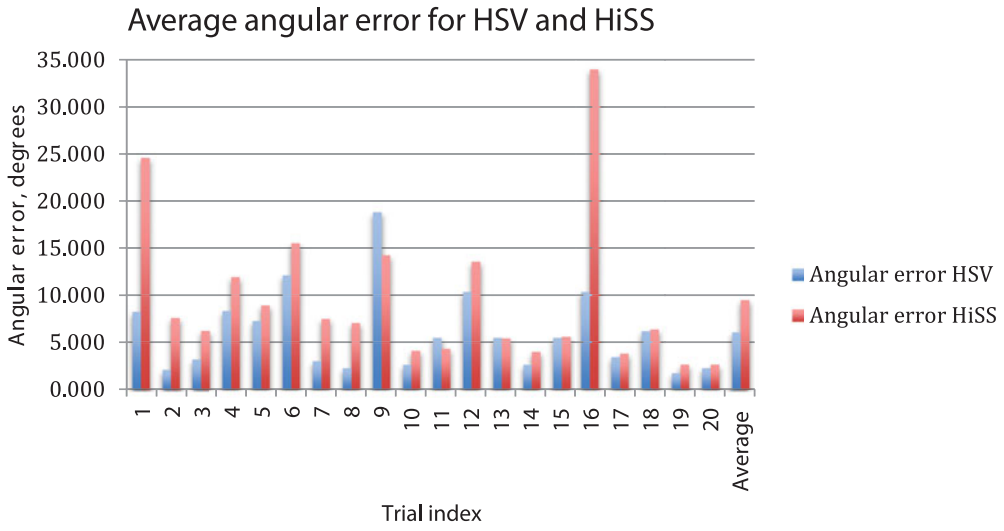


Fig. 12. Comparison of angular error from all trials. Each column represents the average of 5 runs for each trial. The last column is the average over all 20 trials.

The set $\wp$ defines both the start positions and goal positions. For each goal position, trials were conducted on four start positions. Thus, a total of $5 \times 4 \times 5 = 100$ experimental runs were conducted for each algorithm, resulting in a total of 200 runs. For convenience of reference, the trials (combination of start and goal position) are indexed as shown in Table I. All the performance measures were collected for each of the twenty trials (each the average of five runs) in Table I for both the HSV algorithm and the HiSS algorithm. Figure 12 shows the resulting average angular error for each trial for both algorithms. Figure 13 shows the average position error. Figure 14 shows the average return ratio, and Fig. 15 shows the number of homing steps. These results are discussed in the next section.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

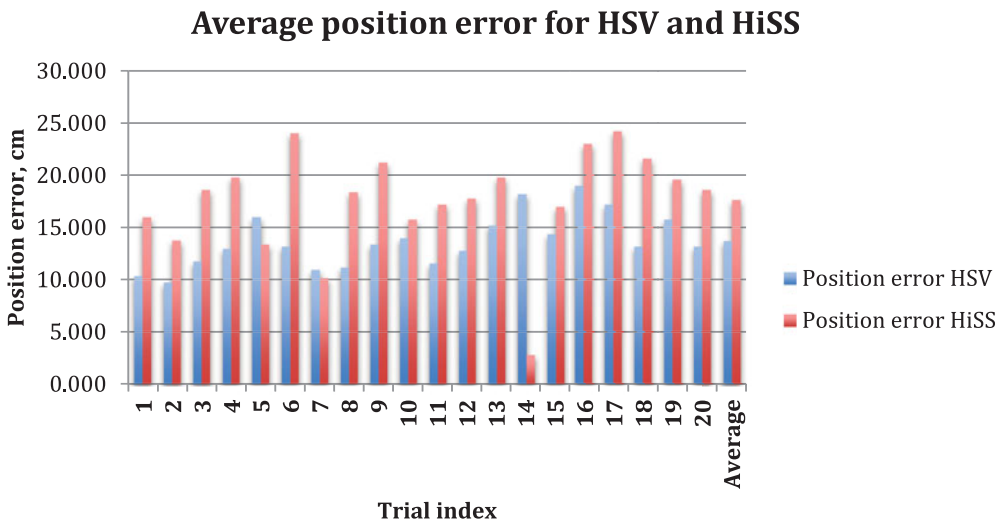


Fig. 13. Comparison of position error for all trials. Each column represents the average of 5 runs.

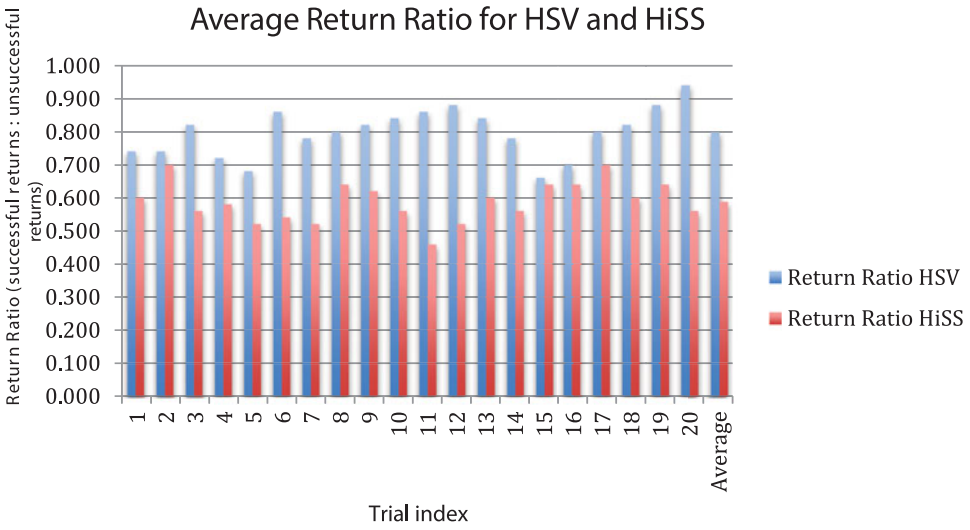


Fig. 14. Comparison of return ratio for all trials. Each column represents the average of runs.

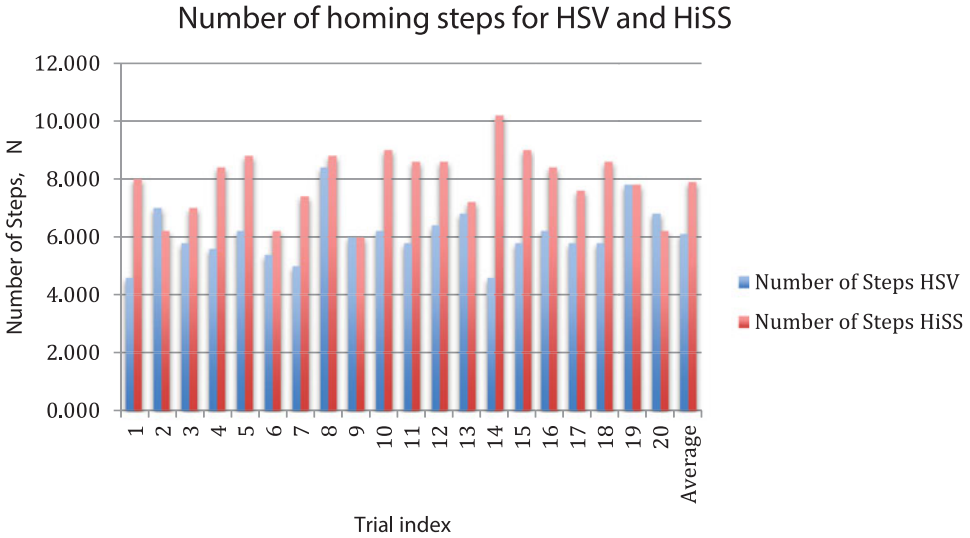


Fig. 15. Comparison of number of steps, *N* from all trials. Each column represents the average of runs.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

6. Discussion

First, we assess the overall significance of these results. Since the experimental tests were conducted at different goal and start locations, it is not safe to assume that the data for angular error, position error, number of steps, and return ratio is normally distributed. Moreover, the data is an overlap of two or more processes, i.e., the data is an aggregate of results obtained from several independent locations. For this reason, the sign test is applied to test the difference between the means of the results from both algorithms rather than the z-test.

For all performance metrics stated in Section 5.1, the sign test reveals a p-value less than the critical p-value of 0.05. This result shows that at least for the range of testing performed in this paper, that HSV exhibits improved performance compared to HiSS for all four performance measures. In the next four subsections, we dig a little deeper into these results for each performance measure.

6.1. Angular error

Figure 12 shows that HSV yields a lower angular error than that of HiSS in the experimental trials conducted. For two locations the HiSS algorithm exhibits an exceptionally high angular error (trial indices 1 and 16). This high angular error could arise because HiSS uses the scale component of the SIFT feature vector to detect contraction and expansion of features between two images. In the presence of camera noise and improper focus, the chance of misclassification between contracted and expanded features may be high.[4]

Based on the evidence of these experiments, HSV exhibits a 36% reduced angular error compared to HiSS. The images returned from the omnidirectional camera used in HiSS are warped. The unwarping process causes a loss of visual quality and increased distortion, which could reduce the chance of features being extracted and matched. This in turn reduces the number of matched features on which the HiSS algorithm can operate. Since the algorithm is heavily dependent on the number of matched features, a reduced number of features affect the returned heading estimate.

The images taken from the stereo camera require no unwarping or other transformations. The images are simply concatenated to produce a wide field image, as explained in Section 2. This results in the image quality being unaltered throughout the algorithm, thus preventing any loss of image quality; therefore the likelihood of SIFT features being matched between two images is not affected.

6.2. Position error

Figure 13 shows that the HSV algorithm position error is 23% less than that of the HiSS algorithm for these experimental trials. Some of this may be due to an inaccurate heading, which results in an inaccurate position, because the final position of the robot is defined by the heading by which the robot rotated. It is worth noting, however, that since visual homing is a technique solely based on vision, the robot can only estimate position based on image comparison which can be a limiting factor in accuracy. Consider the two images in Fig. 16. Although the two images are taken from cameras positioned 20 cm apart from one another, performing HSV or HiSS on 16(a) as the goal image and 16(b) as the current image will return "Goal reached".

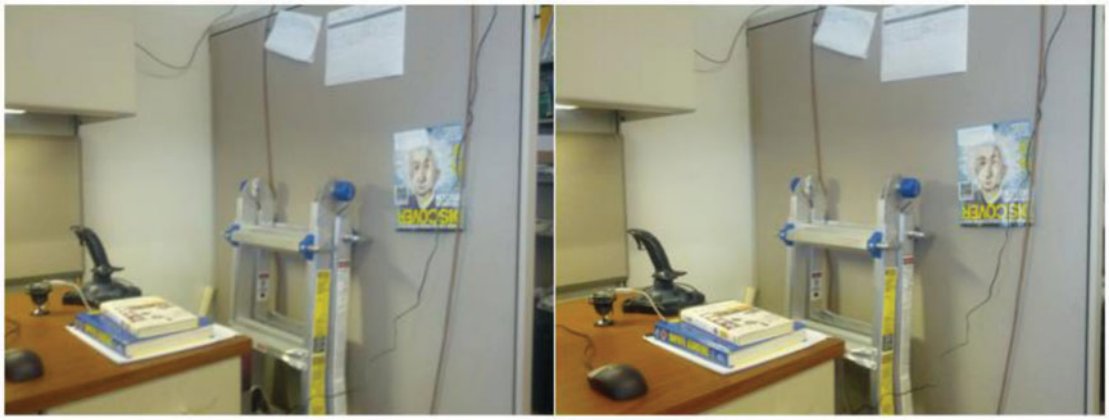
6.3. Number of steps

In these experimental trials, the HSV algorithm requires 23% fewer steps on average to home as shown in Fig. 15. We propose that the main reason behind the reduced number of steps is that the HSV algorithm has more accurate distance estimation than HiSS. However, Churchill & Vardy recently implemented a distance estimation method in a later version of the HiSS algorithm which is based on the percentage of SIFT features matched. The distance d that HiSS, estimates is related to the percentage of SIFT features matched by:

$$d = ae^{bM_{\%}},$$

where $M_{\%}$ is the percentage of SIFT features matched between images C and G. They used nonlinear regression to find the best values for parameters a and b. Future work will include a comparison of HSV with this new approach in HiSS. A more significant set of trials would involve applying distance estimation to the HiSS algorithm, which in turn would generate reduced positional error, and reduced number of steps for the HiSS algorithm.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press



(a) (b)

Fig. 16. Two images taken with the camera positioned 20 cm apart from one another. Both images appear to have little difference, expressing the limitations of image comparison for estimating position.

7. Conclusions

In this paper, we have proposed a novel approach to visual homing for a mobile robot, HSV. The algorithm makes use of SIFT feature matching extended with stereo distance information. The performance of the algorithm was experimentally verified with a set of 200 experimental runs using two Pioneer 3-AT robots. On each trial, the performance of HSV on one robot was compared to that of HiSS[3] which estimates depth from scale information. A set of four performance measures were introduced—angular error, positional error, number of steps, and return ratio—to evaluate the performance of both algorithms. For the 200 robot trials carried out in an indoor, open area with flat surface, our results show that HSV can have significantly improved performance: The angular error exhibited by HSV is significantly less than that of HiSS and the robot required 23% fewer steps to reach the home location, demonstrating effective use of the additional stereo depth accuracy and resolution. However, obtaining the necessary amount of stereo imagery for each homing step required multiple pans of the stereo camera, since the stereo field of view is relatively small. These results should be interpreted also noting that although the same model robot and robot software was used for the HiSS and HSV implementations, different cameras were used. Also, HiSS uses a proportional measurement of error (f^{σ} in Eq. (1)) whereas HSV uses the feature depth directly (f^{z} in Eq. (3)). Since, HSV cannot leverage the proportional approach used by HiSS without a full panorama, the comparison presented therefore is the best that can be achieved with a COTS stereo camera.

Although the experimental trials in this research were conducted at several locations within the test area, the experimental scenario did not change i.e., all trials were conducted in the same indoor, open area environment. A larger set of trials conducted in varying scenarios such as outdoor environments would lead to a more thorough analysis. Using computer simulation can be an effective way to conduct experiments in varied environments. However, it may be difficult to capture some of the issues of noise and uncertainty that surface when conducting experiments with actual robot equipment. Our experimental trials, though conducted in a limited environment, capture the full complexity of real-world sensing and action.

HSV requires several seconds to capture wide-field images due to the time taken by the panning of the stereo camera. HSV could be applied to a robot equipped with a composition of stereo cameras such as Stereo Sphere Vision,[6] thus eliminating the need of a pan-tilt unit and the time taken by panning and tilting the stereo camera. However,[6] only presents a geometric model of Stereo Sphere Vision. A fully implemented Stereo Sphere Vision system is not yet available, but repeating the HSV/HiSS comparison with this equipment would also clarify whether different camera equipment contributed to the performance differences reported in this paper. A Kinect sensor can have a wider field of view depth image since it does not rely on overlapping camera field of views. However, it would still require panning to get a comparable field of view to a panoramic image, and its reliance on IR illumination makes it difficult to work outdoors.

https://doi.org/10.1017/S026357471500034X Published online by Cambridge University Press

Implementing visual homing algorithms in real applications can raise a number of significant challenges such as dynamic obstacles e.g., moving people. Liu *et al*.[10] use visual information coupled with the robot's odometer information to develop an indoor topological navigation framework, resulting in robust performance in real-time environments with scene variation. Ongoing work will include extending and evaluating the robustness of HSV to dynamic obstacles and investigating the extension of the convergence proof to include uncertainty in robot motion.

Acknowledgements
This work was funded in part by the Defense Threat Reduction Agency (DTRA) basic research award # HDTRA1-11-1-0038.

References

1. B. Cartweight and T. Collet, "Landmark learning in bees," *J. Comparative Physiol.* **151**, 521–543 (1983).
2. D. Choi, I. Shim, Y. Bok, T. Oh and I. Kweon, "Autonomous homing based on laser-camera fusion system," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2012).
3. D. Churchill and A. Vardy, "Homing in scale space," *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, (2008).
4. D. Churchill and A. Vardy, "An orientation invarient visual homing algorithm," *J. Intell. Robot. Syst.* **17**(1), 3–29 (2012).
5. G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics* (Cambridge University Press, Cambridge, 2000).
6. W. Feng, B. Zhang, J. Roning, X. Zong and T. Yi, "Panoramic Stereo Vision," *Proceedings of the SPIE Conference on Intelligent Robotics and Computer Vision XXX: Algorithms and Techniques*, Burlingame CA (2013).
7. M. Fischler and R. Bolles, "Random sample concensus: A paradigm for model fitting with applications to image abalysis and automated cartography," *Comm. ACM* **24**(6), 381–395 (1981).
8. M. Franz, B. Scholkopf, M. Mallot and H. Bulthoff, Where did I take that snapshot? Scene-based homing by image matching. *Biol. Cybern.* (79), 191–202 (1998).
9. Y. Jin and M. Xie, "Vision guided Homing for Humanoid Service Robot," *Proceedings of the 15th International Conference on Pattern Recognition (ICPR),* Vol. 4 (2000).
10. M. Liu, C. Pradalier, F. Pomerleau and R. Siegwart, "The Role of Homing in Visual Topological Navigation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2012).
11. D. Lowe, "Distinctive image features from scaleinvarient keypoints," *J. Comput. Vis.* **60**(2), 91–110 (2004).
12. R. Möller, "Do insects use templates or parameters for landmark navigation?," *J. Theor. Biol.* **210**, 33–45 (2001).
13. R. Möller, "Local visual homing by warping of two-dimensional images," *Robot. Auton. Syst.* **31**(1), 87–101 (2009).
14. R. Möller and A. Vardy, "Local visual homing by matched-filter descent in image databases," *Biol. Cybern.* **95**, 413–430 (2006).
15. R. Möller, D. Lambrinos, R. Pfeifer and R. Wehner, "Insect strategies of visual homing in mobile robots," *Comput. Vis. Mobile Robot. Workshop* (1998).
16. P. Nirmal and D. Lyons, "Visual Homing with a Pan-Tilt Based stereo camera," *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision XXX: Algorithms and Techniques*, Burlingame CA (2013).
17. J. Pons, W. Huhner, J. Dahmen and H. Mallot, "Vision-Based Robot Homing in Dynamic Environments," *Proceedings of the 13th IASTED International Conference on Robotics and Applications* (2007).
18. M. Srinivasan, "Insects as gibsonian animals," *Ecol. Psychol.* **10**(3–4), 251–270 (1998).
19. W. Sturzl and H. Mallot, "Vision-Based Homing with a Panoramic Stereovision Sensor," *Proceedings of the British Machine Vision Conference 2002 LNCS 2525* (2002).
20. S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics,* (MIT Press, Cambridge, MA, 2005).
21. A. Vardy and R. Möller, "Biologically plausible visual homing methods based on optical flow techniques," *Connect. Sci.* **17**, 47–90 (2005).
22. A. Vardy and F. Oppacher, "Low-level visual homing," *Advances in artificial life - Proceedings, 7th European Concerence on Artificial Life (vol. 2801 Lecture Notes in Artificial Intelligence)* (2003).
23. R. Wehner, "Spatial Vision in Anthropods," **In:** *Handbook of Sensory Physiology VII/6C, Comparative physiology and evolution of vision in vertebrates* (1981).
24. J. Zeil, H. Hoffman and J. Chal, "Catchment areas of panoramic images in outdoor scenes," *J. Opt. Soc. Am.* **20**(3), 450–469 (2003).