Scale-only Visual Homing from an Omnidirectional Camera

Ming Liu, Cédric Pradalier, François Pomerleau, Roland Siegwart
Autonomous Systems Lab, ETH Zurich, Switzerland
[ming.liu, cedric.pradalier, francois.pomerleau]@mavt.ethz.ch, rsiegwart@ethz.ch

Abstract—Visual Homing is the process by which a mobile robot moves to a Home position using only information extracted from visual data. The approach we present in this paper uses image keypoints (e.g. SIFT) extracted from omnidirectional images and matches the current set of keypoints with the set recorded at the Home location. In this paper, we first formulate three different visual homing problems using uncalibrated omnidirectional camera within the Image Based Visual Servoing (IBVS) framework; then we propose a novel simplified homing approach, which is inspired by IBVS, based only on the scale information of the SIFT features, with its computational cost linear to the number of features.

In addition to simulation results, this paper reports on the application of our method on a commonly cited indoor database where it outperforms other approaches. We also briefly present results on a real robot and allude on the integration into a topological navigation framework.

I. INTRODUCTION

HOMING is defined as the navigation of a robot from an arbitrary position towards a previously specified Home position [1]. In this paper, we consider the case where the control is achieved by extracting visual features and matching them with the features extracted at the home position. Visual Homing is considered to be one of the important abilities of a mobile robot and also one of the most important components of visual topological navigation[2], [3]. In visual topological navigation, the homing method is utilized to perform the transition between topological nodes. Comparing to methods based on metric maps [4], [5], [6], visual topological navigation framework has the following important advantages:

a) Sparse representation of the environment: Usually the topological map used in visual topological navigation is created incrementally, by only considering feature changes. A typical representation of the environment is a collection of visual features at certain poses. The computational and memory cost is relatively low.

b) Independent to precise maps: As a result, visual homing is less sensitive to error accumulation, commonly occurring in metric mapping approaches.

c) Lightweight planning: The path planning in metric maps can be computationally very expensive; in the contrary, the planning of visual topological navigation is based on graph structure with a relatively low number of nodes.

As the key to visual topological navigation, the main challenge in the Visual Homing problem is the estimation of the homing vector, defined as the direction in which the robot has to move to reach the home position. Our method solve this problem by taking inspiration from the generic framework of Visual Servoing and using a omnidirectional camera as the only sensor. “Visual Servoing” [7], [8] has been widely cited in the area of motion control of robotic platforms, such as industrial arms or mobile helicopters. The originality of our approach is that we use and only use the variation of scale information of the SIFT features to compute our simplified control law for visual homing under the image based visual servoing framework. We will show in section IV that the scale of the features provides sufficient information to build an image based visual servoing control law. Since a camera also provides naturally the bearing to the features, we will also show how this information can be used in the control, only bearing and only scale.

In general, visual servoing approaches require the computation of the pseudo-inverse of a matrix whose size is proportional to number of features. For systems with limited resources, this can quickly become intractable. To alleviate that, our approach implements the homing task with a cost linear in the number of features. At the same time, our approach doesn’t depend on calibration parameters. We show in section V that the resulting control is stable and is able to converges exponentially.

The major contributions of this paper are

1) a comparative study of visual homing approaches under IBVS framework for an uncalibrated omnidirectional camera;
2) a robust and fast homing framework based on the scale information of keypoint descriptors, which is independent to the camera calibration;
3) an experimental evaluation of the above on both datasets a mobile robotic platform.

In the following, we first give an overview of related work in Section II. We then formulate the visual homing problem using scale information in the visual servoing framework. Sections V and VI describe the algorithm and control strategy of our approach. Sections VII, VIII and IX include the simulation and experiment results.

This work was supported by the EU FP7 project NIFTi (contract # 247870) and EU project Robots@home (IST-6-045350).
II. RELATED WORK

Visual homing is often implemented using a bearing-only method. An initial work was presented by Cartwright and Colletti [9] as the 'snapshot' model. Franz et al. [10] continued this direction by analyzing the error and convergence properties. In our previous work [11], we gave a proof of the convergence of a simplified bearing-only method, based on Lyapunov stability theory. In this paper, we extend the bearing-only problem to the classical IBVS framework. Some recent work shows that homing can be efficiently achieved with known landmarks [12].

Our novel approach is stimulated by the work of Corke et al. [13], where the authors used the ALV[14] (Average Landmark Vector) principle to implement a visual servoing task. The ALV method converts the homing problem to a vector operation process, by summing up the bearing vectors to a number of keypoints at the reference and current position. The difference between these two sums is then used to compute the homing vector. However this approach depends on knowing the position of the landmarks used as keypoints. This in turns assumes that it is possible to estimate the distance to the keypoints. In comparison our approach takes advantage of the scale information attached to the keypoints to calculate the homing vector without distance estimation and with a computational requirement as low as possible.

According to Goedeme et al. [15], knowing the structure of environment and in particular the landmark position is not necessary in visual homing. These information can be recovered by estimating the ratio of the distances to the matching keypoints by triangulation using an Extend Kalman Filter. Using the features scale, we can avoid this estimation step and use the scale error as a proxy for the distance error.

Among other recent results in the field of homing algorithms, the work of Lim et al[16] divides the 2D plane into four regions and estimates the current robot position by measuring the bearings of landmarks. The theory was proved geometrically but the approach requires more test in a dynamic environment. Cherubini et al [17] proposed a redundancy framework for visual homing problem, which allows online obstacle avoidance.

Note that there are many other works aimed at the visual homing problem, but using different strategies such as [18] and later [19], which relied on the 1D trifocal tensor from the omnidirectional camera. Further, [20] used a sliding-mode control law to exploit the epipolar geometry; [21] directly calculates the homographies from raw images and so on. The comparison with these works is not considered in this report, since the basic strategies and premises are significantly different.

Some related early work using SIFT as main features for visual homing was proposed in [22] [23]. They considered the epipolar geometries as well as the orientation and scale of SIFT features for monocular cameras, following the similar framework proposed in [7]. The work of Vardy et al. [24] is the closest to our simplified approach using scale information.

Their first work developed a scale invariant local image descriptor for visual homing, based on the optical flow of unwrapped panoramic image from an omnidirectional camera. This work was continued by Churchill et al. in [25], which presents results of real-time homing experiment using the scale difference field in panoramic images, computed from SIFT matches. In comparison to their work, we stress the following two main differences: firstly, we reason on the effect of the change of scales in a more dedicated way, by embedding the scale measures inside visual servoing framework. Secondly, we give a mathematical proof of the convergence of the controller. We will refer to their method as “scale space homing” in the following and show our approach outperforms theirs in terms of precision.

III. PROBLEM DEFINITION

A. Formulation

The visual homing problem can be defined as shown in figure 1, where \( p_1, p_2, \ldots, p_n \) are \( n \) keypoints, which are extracted by SIFT, SURF [26] or other method providing the scale information of these keypoints. It is assumed that all the listed keypoints can be seen from the current position \( C \) and the home position \( O \). The objective is to guide the robot from \( C \) to \( O \) only by knowing the observed scale \( s_i \) and bearing angle \( \beta_i \) associated with keypoint \( p_i \). Negre et al. [27] showed that the intrinsic scale can be a measurement to the time to collision. [23] showed a direct relation between the scale and the distance to the feature point. However, for different setup and different environment, the absolute distances to the features cannot be mapped directly. We believe that the variation of the scale of a keypoint can be seen, in first approximation, as a proxy for the variation of its distance.

One fundamental reason is that the scale of a keypoint, in pixels, corresponds to the physical scale of some object imaged from the camera. Using a standard pin-hole model of the camera, the scale is actually inversely proportional to the distance between the camera and the imaged object, assuming the object scale and focal length are constants. As a result, the error between the observed scale and the reference scale gives us an indication of the error between the respective distances and can be used to control the robot towards the home position.

![Fig. 1. Abstracted problem of homing.](image-url)
B. Principles of Visual Servoing

In this paper, we assume that the robot can be controlled using a velocity vector, including directions and absolute values of the speed. This neglects the non-holonomic properties of most robotic platforms. It is acceptable for the simple differential robots used in our experiments, but more work would be needed to adapt this control framework to a more complex system such as a car or a space rover.

When we consider the homing problem as a control problem in the appearance space, it can be summarised to an IBVS (Image Based Visual Servoing) problem. In this context, the objective is to drive an error \( e \) between the observed and desired parameter to zero. In a classical visual servoing approach, the error would be the difference in feature coordinates (in pixel).

According to the fundamental of “Visual Servoing”, the error can be minimised providing the error dynamics can be linked to the control input \( v \) using an interaction matrix \( L_e \) and the following relation:

\[
\dot{e} = L_e v \tag{1}
\]

Once we have calculated the error, we set up a direct P-controller for the motion control of the robot. The controller is designed to eliminate or minimize the error \( e \) with \( v = -\lambda L_e^+ e \), where \( L_e^+ \) is the pseudo-inverse of the interaction matrix \( L_e \). This guarantees the convergence of the error to zero.

IV. IMAGE BASED VISUAL SERVOING

In this paper, we intend to use the visual servoing framework to build a robot controller taking advantage of the scale and bearing to the keypoints, instead of their coordinates. In this section, we introduce how could the interaction matrix \( L_e \) be designed using these features for panoramic images.

We assume that the keypoints are extracted from an omnidirectional camera and we can convert image coordinates to bearing angles. We also assume that we are controlling a ground robot whose configuration can be summarised by its position \((x, y)\) and its heading \(\theta\).

A. Definitions

The error of the system is made of two components: the scale errors and the bearing angle error. Therefore the vector of the error can be written as:

\[
e = (s - s^*, \beta - \beta^*)^T \tag{2}
\]

where \( s = (s_1, \ldots, s_n) \) is the vector of observed scale of the keypoints and \( \beta = (\beta_1, \ldots, \beta_n) \) is the vector of their bearing angles. The variables with \( \ast \) superscripts are reference values.

Before computing the derivative of the error, we need to derive some relations between the scale of a feature \( s_i \) and the distance to the corresponding object \( l_i \). Let us denote \( f \) the focal length of the camera\(^1\) and \( S \) the physical scale of the keypoint. Using simple triangulation and the camera pin-hole model, we have

\[
s_i = \frac{S f}{l_i} \quad \text{and} \quad s_i^* = \frac{S f}{l_i^*} \tag{3}
\]

which leads to

\[
s_i = s_i^* \frac{l_i^*}{l_i} \tag{4}
\]

If we assume that the physical keypoint \( i \) is at the 2D coordinates \((x_i, y_i)\) in the same frame as the robot coordinates, we can also explicit the relation between \((l_i, \beta_i)\) and the robot coordinates:

\[
l_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{5}
\]

\[
\beta_i = \text{atan2}(y_i - y, x_i - x) - \theta \tag{6}
\]

B. Error derivative

To compute the error derive \( \dot{e} \), we derive independently the scale and bearing derivative, by considering them as function of the robot pose. Using equation 4 and 5,

\[
\frac{d}{dt} [s_i(x, y, \theta) - s_i^*] = s_i^* l_i^* [v_x \frac{1}{l_i} + v_y \frac{1}{l_i}]
\]

\[
= -s_i^* l_i^* [v_x \cos \beta_i + v_y \sin \beta_i] \tag{7}
\]

with \( v_x = \frac{dx}{dt} \) and \( v_y = \frac{dy}{dt} \).

Similarly, the bearing error can be derived from equation 6 as:

\[
\frac{d}{dt} [\beta_i(x, y, \theta) - \beta_i^*] = \frac{-y_i - y}{l_i^2} v_x - \frac{x_i - x}{l_i^2} v_y - \omega
\]

\[
- \frac{1}{l_i} [v_x \sin \beta_i + v_y \cos \beta_i] - \beta_i^* \tag{8}
\]

with \( \omega = \frac{d\theta}{dt} \).

Combining equations 7 and 8, we can write the error dynamics as follows:

\[
\frac{d}{dt} e = L_e v \tag{9}
\]

\[
\begin{pmatrix}
s_1 - s_1^* \\
\vdots \\
\beta_1 - \beta_1^* \\
\vdots \\
\beta_n - \beta_n^*
\end{pmatrix}
= \begin{pmatrix}
\frac{-s_1^* l_1^*}{l_1} \cos \beta_1 - \frac{s_1^* l_1^*}{l_1} \sin \beta_1 \\
\vdots \\
\frac{-s_n^* l_n^*}{l_n} \cos \beta_n + \frac{s_n^* l_n^*}{l_n} \sin \beta_n \\
\frac{\beta_1 l_1}{l_1} \cos \beta_1 - \frac{\beta_1 l_1}{l_1} \sin \beta_1 - 1 \\
\vdots \\
\frac{\beta_n l_n}{l_n} \cos \beta_n - \frac{\beta_n l_n}{l_n} \sin \beta_n - 1
\end{pmatrix}
\begin{pmatrix}
v_x \\
v_y \\
\omega
\end{pmatrix}
\]

As mentioned earlier, the interaction matrix in equation 9 is enough to implement a visual servoing controller. One remaining problem is neither the distances \( l_i \) nor \( l_i^* \) can be quantified easily using a single camera. Based on [28] and the analysis on the errors in [8], these values can be approximated by constants due to the low sensitivity of the controller to these parameters. The validity of this assumption will be shown in the simulation in section VII.

\(^1\)In the case of an unwarped catadioptric image, we abuse the notation by relating the focal length to the vertical field of view \(\alpha\) and the image height \(h: \frac{h}{2} = \text{atan2}(h, f)\).
For a real deployment of this algorithm on an embedded controller, it is necessary to note that computing the pseudo-inverse $\mathbf{L}_e^+$ is an expensive operation using standard algorithms. The complexity is cubic with the number of features $n$, which can easily reach several hundreds in typical indoor scenes. Furthermore, this inversion cannot be pre-computed as the matrix depends on the current value of the bearing angles.

A direct way to reduce the complexity is to notice that either the upper part or the lower part of equation 9 are enough to implement a visual servoing task. As it is trivial to rotate the robot on the spot once the translational error has been corrected, a two-stage controller can be considered. We consider only the first stage here - the translation to the home position, because it is the key issue for homing. In practice, this means that we can either implement a scale-only visual servoing or a bearing-only visual servoing.

The interaction matrix for scale-only visual servoing is shown as follows:

$$\frac{d}{dt} \begin{pmatrix} s_1 - s_1^* \\ \vdots \\ s_n - s_n^* \end{pmatrix} = \begin{pmatrix} \alpha_1 s_1^* \cos \beta_1 & \alpha_1 s_1^* \sin \beta_1 \\ \vdots & \vdots \\ \alpha_n s_n^* \cos \beta_n & \alpha_n s_n^* \sin \beta_n \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

(10)

A similar method which uses the lower part of equation 9 is the bearing-only approach. The error dynamics can be derived as:

$$\frac{d}{dt} \begin{pmatrix} \beta_1 - \beta_1^* \\ \vdots \\ \beta_n - \beta_n^* \end{pmatrix} = \begin{pmatrix} \gamma_1 \sin \beta_1 & \gamma_1 \cos \beta_1 \\ \vdots & \vdots \\ \gamma_n \sin \beta_n & \gamma_n \cos \beta_n \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

(11)

For a large number of feature, computing the pseudo-inverse of this matrix would still be computationally expensive. To alleviate this issue, we propose a simplified control approach in the next section, inspired from the scale-only visual servoing, but with a linear complexity. All these approaches will be compared in sections VII and VIII.

V. Fast Visual Homing

In this section, we will describe a scale-based visual homing approach that does not require the computation of the pseudo-inverse of an interaction matrix. We will then prove that the resulting control law converges to the Home position. [23] provided an indication that how the scale of visual features relate to the distance to the certain physical feature point. It stimulated us the possibility to perform homing via only scale information.

A. Scale-based Control for a 1-D Robot

Recall equation 7:

$$\frac{d}{dt} (s_i - s_i^*) = -\frac{s_i^* l_i^*}{l_i^2} [v_x \cos \beta_i + v_y \sin \beta_i]$$

(12)

For the sake of the argument, let us consider a 1-D robot, only able to move towards keypoint $i$. Because the right side of the above equation can be seen as the projection of the robot velocity on direction towards the keypoint, naming $e_i = s_i - s_i^*$ and $v_i = v_x \cos \beta_i + v_y \sin \beta_i$, we have:

$$\frac{d}{dt} e_i = -\frac{s_i^* l_i^*}{l_i^2} v_i$$

(13)

Following the basic strategy of visual servoing, we would like to ensure an exponential decoupled decrease of the error [29]. The following trivial control would achieve this goal ($\lambda$ is a positive constant).

$$v_i = \lambda_i e_i$$

(14)

B. Generic Scale-based Control

If we now, come back to the 2-D case, we can decide to intuitively combine the velocities that would be given by the individual 1-D controllers:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \sum_{i=1}^{n} \lambda_i (s_i - s_i^*) \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix}$$

(15)

However, even if the convergence was obvious in the 1-D case, there is no guarantee that this sum of control contributions would lead to a stable controller. To prove this, we will resort to the Lyapunov theory. Let us define the following non-negative energy function (Lyapunov candidate function):

$$E = \frac{1}{2} \sum_{i=1}^{n} \left( \frac{l_i^2}{s_i^* l_i^*} (s_i - s_i^*)^2 \right)$$

(16)

In this autonomous system with $n$-dimensional states $s$, the only still point is where $s = s^*$ in the feature space; and physically it’s the home position. According to Lyapunov theory, we need to show that,

$$\frac{d}{dt} E(t) \leq 0$$

Based on the calculation in Eq. 7, the derivative of the energy function is:

$$\frac{dE}{dt} = \sum_{i=1}^{n} \frac{s_i - s_i^*}{s_i^*} \frac{ds_i}{dt}$$

$$= -\sum_{i=1}^{n} \frac{s_i - s_i^*}{s_i^*} \frac{l_i^2}{l_i^*} [v_x \cos \beta_i + v_y \sin \beta_i]$$

(17)

We plug in the control law of robot for 2-D plane in equation 14 at this step, the derivative is formulated as:

$$\frac{dE}{dt} = -\left[ v_x \sum_{i=1}^{n} \frac{s_i^* l_i^*}{l_i^2} s_i - s_i^* \cos \beta_i + v_y \sum_{i=1}^{n} \frac{s_i^* l_i^*}{l_i^2} s_i - s_i^* \sin \beta_i \right]$$

By setting

$$\lambda_i = \frac{l_i^*}{l_i^2},$$

(18)

the equation above simplifies to

$$\frac{dE}{dt} = -\left[ v_x^2 + v_y^2 \right] \leq 0.$$
\( \lambda_i \) given in equation 19 is stable and converges to \( s_i = s_i^* \), which is the Home position.

Comparing to [25], we not only manage to calculate a more precise homing direction, the amplitude of the velocity is also indicated.

In practice, \( l_i \) and \( l_i^* \) are not known, but based on [28] and the analysis on the errors in [8], the \( \lambda_i \)'s can be approximated by a constant due to the low sensitivity of the controller to these parameters.

The intuition behind this concept is that each feature point provides a direction and amplitude which the robot should follow, then the output of the controller will be a joined decision of all the observed features. As the number of features is big, even if some of the features cannot be matched with the reference, the performance and convergence of the controller is not jeopardized.

VI. INTEGRATION AND CONTROL LOOP

In this section, we will discuss how the scale-based visual servoing control described above can be instantiated on a real system.

\[ \text{Fig. 2. The control loop of multi-point homing} \]

The control loop is depicted in figure 2. We first focus on the visual processing pipeline on the right hand side of the dotted line. The motion control will not be introduced in this report, as it is strongly related to the design of the robot. The control loop will start by capturing the panoramic image. During each running cycle, the omnidirectional image acquired from the camera is unwrapped first, using a simple polar-coordinate transformation. Note that this implies a minor assumption on the calibration of the camera, namely that the image center is known, but it is not necessary to know a full model of the catadioptric image formation.

The unwrapped image is then compared with the image shot at the Home position. From the matched keypoints, the visual servoing control law is applied, and a homing vector is computed. This vector is then transmitted to the robot motion controller that will try to implement it while accounting for the non-holonomic constraints. As long as the error \( e \) is above a given threshold, the control loop is executed with new images.

VII. SIMULATION RESULTS

In this section we present a number of simulation results highlighting some properties of the homing approaches described above. It is also important to compare all the image-based homing solutions in the aspects of error distribution and error convergence. The four different controllers we are considering are the following:

1) Bearing-only image servoing only takes the bearing to the features to compute the control (see eq. 11).
2) Scale-only image servoing only takes into account the scale of the features to express the control, even though the bearing plays a role in the interaction matrix.
3) Compound image servoing uses the full interaction matrix, combining bearing and scale error.
4) The simplified approach uses scale but does not require the pseudo-inverse computation (see eq. 14).

The left of figure 3 depicts our simulated environment where visual features (blue stars) are assumed to be located on walls surrounding a ground robot. The four visual servoing trajectories shows that all the control laws converges to the home position in a reasonable manner. The center of the figure shows how the visual servoing error converge over time. As expected all approaches exhibit an exponential convergence. The rate of convergence cannot easily be used to compare the different controllers as it strongly depend on the controller gain. Nevertheless, it is interesting to see that with a dramatic reduction in computation, our simplified scale-based visual servoing is just marginally slower than the complete visual servoing approach. On the right side of this figure, it can be observed how the error field change over the environment. Combining bearing and scale error definitely help shaping the error field into a well-defined potential well.

In addition to this particular example, we also ran statistical evaluation by uniformly sampling starting positions in the environment and regenerating the feature set randomly. Detailed results cannot be presented within this article for lack of space, but the bottom line is that the four controllers can achieve similar control accuracies, with a similar number of iteration (differences can be attributed more to the gain tuning than to the specifics of the algorithms). However, the simplified controller and the bearing-only controller turned out to be slightly more efficient in terms of diverging as little as possible from the straight line motion toward the home position. In the same statistical evaluation, it can be shown that the simplified controller has limited sensitivity to noise on the feature scale but that the feature distribution can strongly influence the travelled path. Finally, we also validated the low influence of the controller to the exact knowledge of the distance to the landmarks and its suitability for a non-holonomic platform.

VIII. RESULTS ON AN INDOOR DATASET

In order to compare our approach with other methods under similar conditions, we tested our approach on a widely cited dataset [30]. The dataset is a collection of omni-images and unwrapped images in an indoor environment, plus the
calibration information and pose information. All the images are 561x81 resolution, the actual intervals between two nearest nodes are 30cm. An instance from the database is shown as figure 4.a.

According to the comparison done in [25], the TAAE (total average angular error) can be an important statistic result when evaluating the homing ability. The overall AAE (average angular error) can be obtained as follows:

$$AAE(ss) = \frac{1}{mn} \sum_{x=1}^{m} \sum_{y=1}^{n} AE(ss, cv_{xy})$$

where $AE$ is the absolute angular error between the actual homing vector and ground truth. The subscripts ss and cv stands for saved scene and current view respectively. For the entire image database $db$, the TAAE (total average angular error) computes the over all average of $AAE(ss)$ as follows:

$$TAAE(db) = \frac{1}{mn} \sum_{x=1}^{m} \sum_{y=1}^{n} AAE(ss_{xy})$$

figure 4.b shows the homing vector computed by each controller over the whole dataset, assuming the homing position is at the center position (5,8). The color of the filled circles shows the different number of matching features. It is interesting to see that our simplified controller exhibits a very clean behaviour pointing towards the home position, even when the
matching result is not high. The AAE for each position of the database in given in figure 4.c.

The generic statistic result of the test is shown as figure 4.d. The blue curve indicates the processing time over 28900 matches (cross matching of 17*10 image array). It shows that the processing time is in average around 2Hz to 7Hz, depending on the resolution level. This is defined as the starting octave level for the image pre-processing. Generally speaking, a resolution level of 0 means the original size of the image, -1 means a up-sample of the original image to its double size etc. The effect of this parameter is trivial for different implementations of feature extractions. Nevertheless, we would like to utilize this mean to easily see the influence of the number of features. Intuitively, a bigger image can provide more features. Note that more than 99% of the computation time is for the SIFT extraction and feature matching. Taking the simplified approach as an instance, we plot the TAAE (total average angular error) under different resolution level with error bars, using the purple curve in figure 4.d.

A detailed analysis on the effect of different resolution levels on AAE, and the computation time are shown in table I. It shows that at a higher resolution (lower resolution level), all methods can work better in general. A primary reason is that the higher resolution leads to more feature points, which provides more constraints for the error correction. Note that the simplified method can provide faster and more consistent results than others. The bearing-only visual servoing approach provides best results at node (5,8), in the sense of low AAE, at the cost of 4 to 10 times computation time than the simplified method. However, this result doesn’t suggest the bearing-only method is the best for all the nodes; according to our further test, the TAAE of the bearing-only method is 11.53 degrees, with a bigger variance than the simplified approach. In this report, we chose to focus on the simplified approach for its lower computation requirements. Further comparisons are not included due to limited space.

In real cases, we used image with resolution level -1, with nearly 4.0 Hz processing speed. Practically it is sufficient for real-time applications. The TAAE of our simplified approach (11.01°), over the entire database, outperforms scale space homing method (12.4°) [25] and warping method (snapshot model) (46.6°) [10].

IX. Homing Experiment

The simplified controller was tested on a differential robot with an omni-cam mounted on the top, above the rotation center. A typical setup of the robot system is shown as figure 5.

The experiment was carried out in an office environment, using an external laser tracker as position ground truth. The Home position was set at the center of the room, and the robot was manually driven to random starting positions. Using the homing process, the robot managed to drive back to the Home position. Due to the limited physical space, this specified experiment covered only a small area of the environment. Recorded trajectories are plotted in figure 6.

X. Conclusion

In this paper, we presented a visual homing framework by visual servoing, based on the scale and/or bearing measurement of popular visual features such as SIFT and SURF. After showing how these measurements could be used in the standard visual servoing framework, we proposed a simplified controller with complexity linear with the number of observed features. We’ve demonstrated the usability of scale-based visual servoing and we have shown that our simplified approach is stable and offer similar performances as the fully-fledged scale-based visual servoing. When tested against a standard dataset, our controller outperformed state-of-the-art approaches in terms of average angular error.

Although space is missing to provide details in this paper, our controller was integrated into a visual navigation system, combining a topological map of the environment and our homing controller to align the robot with each topological node. When deployed in the environment depicted in figure 7,

2Extended information: TAAE of other methods are bearing-only(11.53°), scale-only(13.51°), compound method(15.57°).
the controller could successfully move between randomly selected node in the graph for close to one hour. Based on our test results in real apartments and office environments, it also can be considered as a successful algorithm in practical use-cases.

REFERENCES