

Visual Homing From Scale With an Uncalibrated Omnidirectional Camera

Ming Liu, *Student Member, IEEE*, Cédric Pradalier, *Member, IEEE*, and Roland Siegwart, *Fellow, IEEE*

Abstract—Visual homing enables a mobile robot to move to a reference position using only visual information. The approaches that we present in this paper utilize matched image key points (e.g., scale-invariant feature transform) that are extracted from an omnidirectional camera as inputs. First, we propose three visual homing methods that are based on feature scale, bearing, and the combination of both, under an image-based visual servoing framework. Second, considering computational cost, we propose a simplified homing method which takes an advantage of the scale information of key-point features to compute control commands. The observability and controllability of the algorithm are proved. An outlier rejection algorithm is also introduced and evaluated. The results of all these methods are compared both in simulations and experiments. We report the performance of all related methods on a series of commonly cited indoor datasets, showing the advantages of the proposed method. Furthermore, they are tested on a compact dataset of omnidirectional panoramic images, which is captured under dynamic conditions with ground truth for future research and comparison.

Index Terms—Omnidirectional camera, topological visual navigation, visual homing, visual servoing.

I. INTRODUCTION

VISUAL homing is defined as navigation by vision of a robot from an arbitrary starting position to a previously specified reference home position [2]. It is considered to be one of the basic abilities of a mobile robot, as well as one of the most important components of visual topological navigation [3], [4]. On one hand, visual homing is a lightweight method for robot navigation. It utilizes only one pair of images taken at reference and current poses to navigate a mobile robot, regardless the path and motion before reaching the current state. On the other hand, it can be easily extended to an integrated navigation system, by sequentially setting the target nodes on a graph as references. Compared to methods that are based on metric maps [5]–[7], the visual topological navigation framework has the following advantages.

Manuscript received January 28, 2013; accepted June 26, 2013. Date of publication August 1, 2013; date of current version December 2, 2013. This paper was recommended for publication by Associate Editor E. Marchand and Editor G. Oriolo upon evaluation of the reviewers' comments. This paper was presented in part at the 2012 IEEE International Conference on Robotics and Automation. This work was supported in part by the EU project Robots@home under Grant IST-6-045350 and in part by EU FP7 project NIFTi under Contract 247870.

M. Liu is with Autonomous Systems Lab, ETH Zürich, Zürich 8092, Switzerland and also with The Hong Kong University of Science and Technology, Hong Kong (e-mail: ming.liu@mavt.ethz.ch).

C. Pradalier and R. Siegwart are with Autonomous Systems Lab, ETH Zürich, Zürich 8092, Switzerland (e-mail: cedric.pradalier@georgiatech-metz.fr; rsiegwart@ethz.ch).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2013.2272251

- 1) *Sparse representation of environment*: Usually, the topological map that is used in visual topological navigation is created incrementally, considering the 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.
- 2) *Independent of precise maps*: As a result, visual homing is less sensitive to error accumulation, which commonly occurs in metric approaches.
- 3) *Lightweight planning*: The path planning on metric maps can be computationally very expensive; conversely, the planning of visual topological navigation is based on graph structures, which is easy to compute.

The primary challenge of the visual homing problem is the estimation of the *homing vector*. It defines the direction along which the robot can reach the home position. Our methods solve this problem by taking inspiration from the generic framework of visual servoing [8], [9]. Visual servoing has been widely applied in the area of motion control, such as the control of robotic arms for industrial robots. In this paper, we propose a generic visual servoing framework for the visual homing problem. This is implemented as three different homing methods using different properties of features. When deploying such an approach on real robots, special care must be taken with the computational complexity of the algorithm. In general, visual servoing approaches require the computation of the pseudoinverse of a matrix whose size is proportional to the number of sampled features n . In order to ameliorate this, we propose an approach that is inspired from the visual servoing approach, but with a cost linear in the number of features. We show in Section VI that the resulting control is stable, and we compare its performance with other related methods. The originality of this paper is that we take advantage of the scale information of the scale-invariant feature transform (SIFT) features to compute our control law. We show in Section IV that the scale of the features is sufficient to build such a control law.

For practical implementations of visual homing methods, the robotic platforms necessarily have to manage the maneuvers in the presence of dynamic objects, i.e., objects whose position may change from the reference to the current image. If not accounted for, features on these objects will be integrated in the control output, and may lead to unstable or unsafe behavior of the robot. A fast outlier rejection approach is discussed in this paper. It is designed to handle such problems and to improve the overall system performance. The parametrization and validation of the approach will be presented in simulation and experiment.

The major contributions of this paper are as follows.

- 1) A comparative study of three novel visual homing approaches that are based on bearing and scale information

under classical image-based visual servoing (IBVS) framework.

- 2) Observability analysis on the system states regarding navigation problem for an omnidirectional camera.
- 3) A robust and fast homing method that is based on scale information of key-point descriptors, with the proof of stability.
- 4) Evaluation of the aforementioned methods, as well as an outlier rejection method, via simulations and experiments.
- 5) A concise dataset for future study of visual control problems using omnidirectional cameras.

In the following, we first give an overview of related work in Section II. Section III reviews the visual homing problem under visual servoing framework. Sections IV and V discuss the fundamental aspects and observability of visual servoing-based homing. Section VI describes the algorithm and control strategy of a simplified homing algorithm. The control loop is introduced in Section VII, followed by the simulation and experimental results. Conclusions and our future visions are provided at the end of the paper.

II. RELATED WORK

Visual homing is often implemented using bearing-only methods. An initial work was presented by Cartwright and Collett [10] as the “snapshot” model. Franz *et al.* [11] continued this direction by analyzing the error and convergence properties. In our previous work [12], we gave a proof of the convergence of a simplified bearing-only method, based on the Lyapunov stability theory. In this paper, we formulate the bearing-only problem [12] as part of the classical IBVS framework. Meanwhile, two other homing methods under this framework are discussed as well.

Our work is stimulated by the work of Corke [13], where the author used the average landmark vector (ALV) [14] 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 key points at the reference and the current positions. The difference between these two sums is then used to compute the homing vector. However, the following limitations should be considered. First, outliers in landmark detection will greatly affect the accuracy of the homing vector. Second, the outputting homing vector of the ALV algorithm highly depends on the estimation of the global orientation. Third, the homing vector is directly calculated from summarized bearing information of landmarks in current robot pose frame, while the contribution of each landmark is transparent to the final output. Finally, the algorithm is heuristic at intermediate poses. This means that the calculation of accurate homing direction is not feasible, and the mathematical convergence of the algorithm is not guaranteed. In comparison, our approach takes advantage of the scale information attached to the key points to calculate the homing vector without distance estimation, the usage of scale information guarantees the convergence and observability of the system states. The proofs are conducted in this paper.

According to Goedeme *et al.* [15], knowing the structure of an environment and in particular the landmark positions is not

necessary for visual homing. This information can be recovered by estimating the ratio of the distances to the matching key points by triangulation using an extended Kalman Filter. Using feature scales, we can avoid this estimation phase and use the scale variation as a proxy for distance errors.

Lim and Barnes [16] presented a homing algorithm that was based on the following principle. They divided the 2-D plane into four regions and estimated the current robot position by measuring the bearings to known landmarks. Compared with their approach, we prove the convergence of our method using the Lyapunov theory. It guarantees the stability of the controller.

A new trend for solving visual homing problems [17] was proposed by Aranda *et al.* and improves the performances by using 1-D trifocal tensors from the omnidirectional camera. Compared with the works using 1-D-trifocal tensors [18], [19], which rely on three-view geometry, our method infers the homing vector directly from the current appearance, and the result is less reliant on feature association. Besides, our approach does not require solving nonlinear equations constructed from the tensors, where algebra error is embedded in the SVD process and impossible to eliminate. Since the reconstruction of the environmental structure is not needed, our servoing-based method requires less computational power.

Furthermore, the authors in [20] used a sliding-mode control law to exploit the epipolar geometry; the authors in [21], directly calculated the homographies from raw images; and Cherubini and Chaumette [22] proposed a redundant framework for visual homing problem, which in particular allows online obstacle avoidance. The comparison with these works is not considered here, since the basic strategies and premises are significantly different.

Some related early works used SIFT as main features for visual homing [23], [24]. They considered the epipolar geometries as well as the orientation and scale of SIFT features for monocular cameras, following a framework similar to [8]. Among these, the work by Vardy and Oppacher [25] is the closest to our simplified approach using scale information. Their work developed a scale invariant local image descriptor for visual homing, based on the optical flow of unwrapped panoramic image from an omnidirectional camera. It was continued by Churchill *et al.* [26], 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. First, we utilize the error that is caused by the variation of scales, by embedding the scale measures inside the visual servoing framework. Second, we give a mathematical proof of the convergence of the controller, and show the observability of feature states and robot headings. We refer to their method as HSVS, namely heuristic scale-space visual servoing, in the remainder of the paper.

III. PROBLEM DEFINITION

A. Formulation

The visual homing problem can be defined as shown in Fig. 1, where $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ are n key points, which are extracted by SIFT, SURF [27], or other methods providing the scale

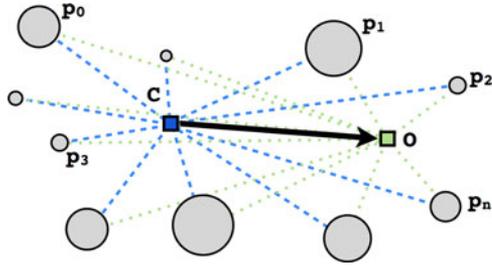


Fig. 1. Abstracted problem of homing. Key points p_1 to p_n can be observed at the current position C and at the Home position O . The variant sizes indicate the differences of key points in scale.

information of key points. It is assumed that all the listed key points 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 β_i that are associated with key point p_i . Negre *et al.* [28] showed that the intrinsic scale can be used to aid the measurement of the time to collision. Hoffmann *et al.* [24] 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. The scale information that we extract from the key points comes from the interpolation of the difference-of-Gaussian pyramid levels, which is commonly used as SIFT key-point detector. We believe that the variation of the scale of a key point can be seen, in first approximation, as a proxy for the variation of its distance. Furthermore, by introducing the scale information, we also prove that the distances to key points are observable given the control speed, as discussed in Section V.

B. Principles of Visual Servoing

In this paper, we assume that the robot can be controlled using a velocity vector, including directions and the absolute values of the speed. This neglects the nonholonomic 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 summarized as an IBVS problem. In this context, the objective is to drive an error e between the observed and desired features to zero. In a classical visual servoing approach, the error would be the difference in feature coordinates (in pixels). According to the fundamentals of visual servoing, this error can be minimized, if the error dynamics can be linked to the control input v using an *interaction matrix* L_e and the following relation [29]:

$$\dot{e} = L_e v. \quad (1)$$

Once the visual error can be properly represented, direct control commands can be calculated in order to minimize the error e . Generally, the control commands can be represented in the form

$$v = -\lambda L_e^+ e \quad (2)$$

where L_e^+ is the pseudoinverse of the *interaction matrix* L_e . This controller is designed to have an exponential convergence rate of the error, if the stability of the system is ensured. According to the stability analysis of IBVS in [29], the interaction matrix and its pseudoinverse need to be full rank, in order to guarantee local asymptotic stability.

IV. IMAGE-BASED VISUAL SERVOING

In this paper, we adopt the visual servoing framework to build a robot controller, by using the scale and bearing information of the key points, instead of their coordinates. We assume that the key points are extracted from an omnidirectional camera, and we can easily convert image coordinates to bearing angles. We also assume that we are able to control the target robot by velocity commands, and the robot configuration can be summarized by its position (x, y) and its heading θ .

A. Definitions

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

$$e = (s - s^*, \beta - \beta^*)^T \quad (3)$$

where $s = (s_1, \dots, s_n)$ is the vector of observed scale of the key points, and $\beta = (\beta_1, \dots, \beta_n)$ is the vector of their bearing angles. The “*” superscript denotes the reference variables.

Before computing the derivative of the error, we need to derive the relation between the scale of a feature s_i and the distance to the corresponding entity l_i . The focal length of the camera is denoted by f and S denotes the physical size of the region defined by the corresponding key-point patch. Using simple triangulation and the camera pin-hole model, we have

$$s_i = \frac{S f}{l_i}, \text{ and } s_i^* = \frac{S f}{l_i^*} \quad (4)$$

which leads to

$$s_i = s_i^* \frac{l_i^*}{l_i}. \quad (5)$$

Assuming that the physical key point i is located at the 2-D coordinates (x_i, y_i) in the same frame as the robot, we can make explicit the relation between l_i , β_i , and the robot position

$$l_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (6)$$

$$\beta_i = \text{atan2}(y_i - y, x_i - x) - \theta. \quad (7)$$

B. Error Derivative

To derive the error dynamics \dot{e} , we compute independently the scale and bearing derivatives, by considering them as a function of the robot pose

$$\frac{d}{dt} [s_i(x, y, \theta) - s_i^*] = \frac{d s_i}{d x} \frac{d x}{d t} + \frac{d s_i}{d y} \frac{d y}{d t} + \frac{d s_i}{d \theta} \frac{d \theta}{d t}. \quad (8)$$

Using (5) and (6), we have

$$\begin{aligned} \frac{d}{dt} [s_i(x, y, \theta) - s_i^*] &= s_i^* l_i^* \left[v_x \frac{d}{dx} \frac{1}{l_i} + v_y \frac{d}{dy} \frac{1}{l_i} \right] \\ &= -\frac{s_i^* l_i^*}{l_i^2} [v_x \cos \beta_i^g + v_y \sin \beta_i^g] \end{aligned} \quad (9)$$

with $v_x = \frac{dx}{dt}$, and $v_y = \frac{dy}{dt}$.

Similarly, the bearing error can be derived as

$$\frac{d}{dt} [\beta_i^g(x, y, \theta) - \beta_i^*] = \frac{d\beta_i^g}{dx} \frac{dx}{dt} + \frac{d\beta_i^g}{dy} \frac{dy}{dt} + \frac{d\beta_i^g}{d\theta} \frac{d\theta}{dt}. \quad (10)$$

By plugging in the (7), this leads to

$$\begin{aligned} \frac{d}{dt} [\beta_i^g(x, y, \theta) - \beta_i^*] &= -\frac{y_i - y}{l_i^2} v_x - \frac{x_i - x}{l_i^2} v_y - \omega \\ &= -\frac{1}{l_i} [v_x \sin \beta_i^g + v_y \cos \beta_i^g] - \omega \end{aligned} \quad (11)$$

with $\omega = \frac{d\theta}{dt}$.

Similarly, the derivative of the distance to a key point i is

$$\frac{d}{dt} [l_i(x, y, \theta) - l_i^*] = -(v_x \cos \beta_i^g + v_y \sin \beta_i^g). \quad (12)$$

Note that β_i^g is the bearing angle of feature i in the global frame instead of robot local frame. In order to transform them to the robot local frame, the heading difference needs to be considered, such that the bearing observation by the robot is

$$\beta_i = \beta_i^g - (\theta - \theta^*). \quad (13)$$

In practice, we can consider $\theta^* = 0$, meaning that we take the reference robot pose as globally zero degree heading.

Combining (9) to (11), we can write the error dynamics as follows:

$$\begin{aligned} \dot{e} &\doteq \frac{d}{dt} e = L_e v \\ \frac{d}{dt} \begin{pmatrix} s_1 - s_1^* \\ \vdots \\ s_n - s_n^* \\ \beta_1 - \beta_1^* \\ \vdots \\ \beta_n - \beta_n^* \end{pmatrix} &= \begin{pmatrix} -\frac{s_1^* l_1^*}{l_1^2} \cos \beta_1^g & -\frac{s_1^* l_1^*}{l_1^2} \sin \beta_1^g & 0 \\ \vdots & \vdots & \vdots \\ -\frac{s_n^* l_n^*}{l_n^2} \cos \beta_n^g & -\frac{s_n^* l_n^*}{l_n^2} \sin \beta_n^g & 0 \\ -\frac{1}{l_1} \sin \beta_1^g & -\frac{1}{l_1} \cos \beta_1^g & -1 \\ \vdots & \vdots & \vdots \\ -\frac{1}{l_n} \sin \beta_n^g & -\frac{1}{l_n} \cos \beta_n^g & -1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix}. \end{aligned} \quad (14)$$

As mentioned earlier, the interaction matrix in (14) is required for the implementation of a visual servoing controller. One remaining problem is that neither the distance l_i nor l_i^* can be quantified easily using a single camera. Based on [30] and the analysis on the visual errors in [9], we assume these values can be approximated by constants due to the low sensitivity of the controller to these parameters.

A direct way to reduce the complexity is that either the upper part or the lower part of (14) is sufficient to form a visual servoing interaction matrix. As it is trivial to rotate the robot on spot once the translational error has been corrected, a two-stage controller can be designed. First, deal with the translation error, then correct the heading. We consider mainly the first stage, since 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:

$$\frac{d}{dt} \begin{pmatrix} s_1 - s_1^* \\ \vdots \\ s_n - s_n^* \end{pmatrix} = \begin{pmatrix} \alpha_1 s_1^* \cos \beta_1^g & \alpha_1 s_1^* \sin \beta_1^g \\ \vdots & \vdots \\ \alpha_n s_n^* \cos \beta_n^g & \alpha_n s_n^* \sin \beta_n^g \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}. \quad (15)$$

The controller that is based on (15) is denoted as scale-only visual servoing (SOVS) in the remainder of this paper. A similar method, using the lower part of (14), is called a bearing-only approach (BOVS), whose error dynamics can be derived as

$$\frac{d}{dt} \begin{pmatrix} \beta_1 - \beta_1^* \\ \vdots \\ \beta_n - \beta_n^* \end{pmatrix} = \begin{pmatrix} \gamma_1 \sin \beta_1^g & \gamma_1 \cos \beta_1^g \\ \vdots & \vdots \\ \gamma_n \sin \beta_n^g & \gamma_n \cos \beta_n^g \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}. \quad (16)$$

According to the generic properties of the controller that are stated in Section III-B, the local asymptotic stability is maintained if each interaction matrix and its pseudoinverse are full ranked. This can be ensured by using reasonable big number of matched features in real applications.

Recalling (13), the estimation of the heading θ is crucial for the calculation of the interaction matrices. This implies that a robot may need to have absolute heading references such as a magnetic compass or a reliable visual compass for better accuracy. Regarding the dataset that we use in Section IX, where the robot is well aligned, this problem is trivial. However, this matter needs to be considered in real applications.

V. OBSERVABILITY OF THE SCALE-BASED VISUAL CONTROL SYSTEM

A. Definitions

Given the raw sensor measurements, the observability analysis of system states is important before we step on the designation of other advanced controllers. The observability analysis also gives hints for what results can be expected from the system configuration. The configuration of the discussed IBVS system is described via the following system states:

$$\mathbf{x} = \begin{cases} x : x\text{-coordinate of the robot position} \\ y : y\text{-coordinate of the robot position} \\ \theta : \text{heading of the robot} \\ \mathbf{s} : \text{vector of observed scales} \\ \boldsymbol{\beta} : \text{vector of observed bearings} \\ \mathbf{l} : \text{distance to features.} \end{cases} \quad (17)$$

In order to get the full state for a motion constrained in 2-D plane, we require at least three positively matched key points, considering the number of geometrical constraints. Without the

loss of generality, we can use the following subset of the full state, where the minimum required key points are denoted by subscripts 1, 2, and 3

$$\underline{\mathbf{x}} = (x \ y \ \theta \ s_1 \ s_2 \ s_3 \ \beta_1 \ \beta_2 \ \beta_3 \ l_1 \ l_2 \ l_3)^T. \quad (18)$$

Following the definitions in (9), (11), and (12), using “*” to denote the reference variables, the state derivatives of the system state is presented as (19), shown at the bottom of the page.

The observation function, which is also the zero-order Lie derivative, includes the scale and bearing information

$$L^0 h = h(x) = (s_1 \ s_2 \ s_3 \ \beta_1 \ \beta_2 \ \beta_3)^T. \quad (20)$$

The further Lie derivatives of the observation function over the control functions are shown in (21)–(23). The control functions are comprised of the three columns of (19), denoted by f_1, f_2, f_3 , respectively. For the purpose of conciseness, we use C_i to denote $\cos(\beta_i + \theta - \theta^*)$, and S_i for $\sin(-\beta_i - \theta + \theta^*)$

$$\begin{aligned} \nabla L^0 h &= \nabla h(x) \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \quad (21) \end{aligned}$$

B. Observation Matrix

We stop the computation by (23), because it has illustrated that the observation matrix shown in (24) is already row full ranked using (21)–(23), but not column full ranked. We notice that continuing to calculate further derivatives would not help it become column full rank, since the entries are independent of x or y , (22)–(24), all shown at the bottom of the next page.

Using all the measurable variables and the known control commands, the analysis of the null space of M reveals the inherent relations among each state. The null space of the full observation matrix M is given as

$$\text{nullspace}(M) = \{[1, 0, 0, \mathbf{0}, \mathbf{0}, \mathbf{0}], [0, 1, 0, \mathbf{0}, \mathbf{0}, \mathbf{0}]\}. \quad (25)$$

This shows that the position of the robot (x, y) in the global frame is not observable due to nonzero entries in the nullspace. This is fine, since they are not of interest without knowing the global reference frame. More importantly, it shows that the other states θ, s, β , and particularly the distances to the feature points l are observable, as long as the control command to the robot is known.

Though not explicitly calculated by the homing algorithm, these observable constraints imply that it is able to determine the translation and rotation to a predefined position, since three known distances to static feature points are adequate to well determine such a transform for the motion in a 2-D plane.

VI. FAST VISUAL HOMING

From the observability analysis, we could see that the scale information and bearing angles from local frame and especially their changes are sufficient information for pose control. Derived from this, we describe a scale-based visual homing approach that does not require the computation of the pseudoinverse of an interaction matrix in this section. Above all, this approach is independent of the global heading estimation. Since the global heading is usually approximated by visual compass and with nonneglectable error [31], this method avoids such potentially extra errors for real applications. We also provide the convergence proof for the resulting control law.

$$\dot{\underline{\mathbf{x}}} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ \sin \theta & -\cos \theta & 0 \\ 0 & 0 & 1 \\ -\frac{s_1^* l_1^* \cos(\beta_1 + (\theta - \theta^*))}{l_1^2} & -\frac{s_1^* l_1^* \sin(\beta_1 + (\theta - \theta^*))}{l_1^2} & 0 \\ -\frac{s_2^* l_2^* \cos(\beta_2 + (\theta - \theta^*))}{l_2^2} & -\frac{s_2^* l_2^* \sin(\beta_2 + (\theta - \theta^*))}{l_2^2} & 0 \\ -\frac{s_3^* l_3^* \cos(\beta_3 + (\theta - \theta^*))}{l_3^2} & -\frac{s_3^* l_3^* \sin(\beta_3 + (\theta - \theta^*))}{l_3^2} & 0 \\ -\frac{\sin(\beta_1 + (\theta - \theta^*))}{l_1} & -\frac{\cos(\beta_1 + (\theta - \theta^*))}{l_1} & -1 \\ -\frac{\sin(\beta_2 + (\theta - \theta^*))}{l_2} & -\frac{\cos(\beta_2 + (\theta - \theta^*))}{l_2} & -1 \\ -\frac{\sin(\beta_3 + (\theta - \theta^*))}{l_3} & -\frac{\cos(\beta_3 + (\theta - \theta^*))}{l_3} & -1 \\ -\cos(\beta_1 + (\theta - \theta^*)) & -\sin(\beta_1 + (\theta - \theta^*)) & 0 \\ -\cos(\beta_2 + (\theta - \theta^*)) & -\sin(\beta_2 + (\theta - \theta^*)) & 0 \\ -\cos(\beta_3 + (\theta - \theta^*)) & -\sin(\beta_3 + (\theta - \theta^*)) & 0 \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix}. \quad (19)$$

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

Recalling (9)

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

For the sake of argument, let us first consider an 1-D robot, which is only able to move along the direction toward key point i . Because the right side of the aforementioned equation can be seen as the projection of the robot velocity in the direction toward the key point, denoting $e_i = s_i - s_i^*$, and $v_i = v_x \cos \beta_i + v_y \sin \beta_i$, we have

$$\frac{d}{dt}e_i = -\frac{s_i^* l_i^*}{l_i^2} v_i. \quad (26)$$

Following the designation 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 (λ is

a positive constant)

$$v_i = \lambda_i e_i. \quad (27)$$

B. Generic Scale-Based Control

We abuse the intuition that the individual controllers for the 1-D case may be combined to calculate the required velocity for the 2-D case as follows:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \sum_{i=1}^n \lambda_i (s_i - s_i^*) \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix}. \quad (28)$$

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. In order to show the convergence, we resort to the Lyapunov theory. We define the following

$$L_{f_1}^1 h = \frac{\partial h(x)}{\partial x} \cdot f_1 = \nabla L_{f_1}^0 h \cdot f_1 = \begin{pmatrix} -\frac{s_1^* l_1^* C_1}{l_1^2}, & -\frac{s_2^* l_2^* C_2}{l_2^2}, & -\frac{s_3^* l_3^* C_3}{l_3^2}, & \frac{S_1}{l_1}, & \frac{S_2}{l_2}, & \frac{S_3}{l_3} \end{pmatrix}^T$$

$$\nabla L_{f_1}^1 h = \begin{pmatrix} 0 & 0 & -\frac{s_1^* l_1^* S_1}{l_1^2} & 0 & 0 & 0 & -\frac{s_1^* l_1^* S_1}{l_1^2} & 0 & 0 & 2 \frac{s_1^* l_1^* C_1}{l_1^3} & 0 & 0 \\ 0 & 0 & -\frac{s_2^* l_2^* S_2}{l_2^2} & 0 & 0 & 0 & 0 & -\frac{s_2^* l_2^* S_2}{l_2^2} & 0 & 0 & 2 \frac{s_2^* l_2^* C_2}{l_2^3} & 0 \\ 0 & 0 & -\frac{s_3^* l_3^* S_3}{l_3^2} & 0 & 0 & 0 & 0 & 0 & -\frac{s_3^* l_3^* S_3}{l_3^2} & 0 & 0 & 2 \frac{s_3^* l_3^* C_3}{l_3^3} \\ 0 & 0 & -\frac{C_1}{l_1} & 0 & 0 & 0 & -\frac{C_1}{l_1} & 0 & 0 & -\frac{S_1}{l_1^2} & 0 & 0 \\ 0 & 0 & -\frac{C_2}{l_2} & 0 & 0 & 0 & 0 & -\frac{C_2}{l_2} & 0 & 0 & -\frac{S_2}{l_2^2} & 0 \\ 0 & 0 & -\frac{C_3}{l_3} & 0 & 0 & 0 & 0 & 0 & -\frac{C_3}{l_3} & 0 & 0 & -\frac{S_3}{l_3^2} \end{pmatrix}. \quad (22)$$

$$L_{f_2}^1 h = \frac{\partial h(x)}{\partial x} \cdot f_2 = \nabla L_{f_2}^0 h \cdot f_2 = \begin{pmatrix} -\frac{s_1^* l_1^* S_1}{l_1^2}, & -\frac{s_2^* l_2^* S_2}{l_2^2}, & -\frac{s_3^* l_3^* S_3}{l_3^2}, & -\frac{C_1}{l_1}, & -\frac{C_2}{l_2}, & -\frac{C_3}{l_3} \end{pmatrix}^T$$

$$\nabla L_{f_2}^1 h = \begin{pmatrix} 0 & 0 & -\frac{s_1^* l_1^* C_1}{l_1^2} & 0 & 0 & 0 & -\frac{s_1^* l_1^* C_1}{l_1^2} & 0 & 0 & 2 \frac{s_1^* l_1^* S_1}{l_1^3} & 0 & 0 \\ 0 & 0 & -\frac{s_2^* l_2^* C_2}{l_2^2} & 0 & 0 & 0 & 0 & -\frac{s_2^* l_2^* C_2}{l_2^2} & 0 & 0 & 2 \frac{s_2^* l_2^* S_2}{l_2^3} & 0 \\ 0 & 0 & -\frac{s_3^* l_3^* C_3}{l_3^2} & 0 & 0 & 0 & 0 & 0 & -\frac{s_3^* l_3^* C_3}{l_3^2} & 0 & 0 & 2 \frac{s_3^* l_3^* S_3}{l_3^3} \\ 0 & 0 & -\frac{S_1}{l_1} & 0 & 0 & 0 & -\frac{S_1}{l_1} & 0 & 0 & -\frac{C_1}{l_1^2} & 0 & 0 \\ 0 & 0 & -\frac{S_2}{l_2} & 0 & 0 & 0 & 0 & -\frac{S_2}{l_2} & 0 & 0 & -\frac{C_2}{l_2^2} & 0 \\ 0 & 0 & -\frac{S_3}{l_3} & 0 & 0 & 0 & 0 & 0 & -\frac{S_3}{l_3} & 0 & 0 & -\frac{C_3}{l_3^2} \end{pmatrix}. \quad (23)$$

$$M = \begin{pmatrix} \nabla L^0 h \\ \nabla L_{f_1}^1 h \\ \nabla L_{f_2}^1 h \end{pmatrix} \quad (24)$$

nonnegative energy function (Lyapunov candidate function):

$$E = \frac{1}{2} \sum_{i=1}^n \left(\frac{s_i - s_i^*}{s_i^*} \right)^2. \quad (29)$$

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

$$\begin{cases} \frac{d}{dt}E(t) = 0, & \text{only when all } s_i = s_i^* \\ \frac{d}{dt}E(t) < 0, & \text{otherwise.} \end{cases} \quad (30)$$

Based on the calculation in (9), the derivative of the energy function is

$$\begin{aligned} \frac{dE}{dt} &= \sum_{i=1}^n \frac{s_i - s_i^*}{s_i^*} \frac{ds_i}{dt} \\ &= - \sum_{i=1}^n \frac{s_i - s_i^*}{s_i^*} \frac{s_i^* l_i^*}{l_i^2} [v_x \cos \beta_i + v_y \sin \beta_i] \\ &= - \left[v_x \sum_{i=1}^n \frac{s_i^* l_i^*}{l_i^2} \frac{s_i - s_i^*}{s_i^*} \cos \beta_i + v_y \sum_{i=1}^n \frac{s_i^* l_i^*}{l_i^2} \frac{s_i - s_i^*}{s_i^*} \sin \beta_i \right]. \end{aligned} \quad (31)$$

Denoting

$$\lambda_i = \frac{l_i^*}{l_i^2} \quad (32)$$

and combining with (28), (31) is simplified as

$$\frac{dE}{dt} = - [v_x^2 + v_y^2] = - \sum \lambda_i^2 (s_i - s_i^*)^2 \leq 0. \quad \square \quad (33)$$

Since the distances l_i and l_i^* are nonnegative, (33) shows that the control law of (28) is stable, and converges to $s_i = s_i^*$ (i.e., the reference home position). However, l_i and l_i^* are not directly measurable in practice, though observable. Following the error analysis in [9], we approximate the λ_i 's by constants, since they do not affect the convergence. Further validation via simulation is given in Section VIII-B1).

VII. INTEGRATION AND CONTROL LOOP

In this section, we will discuss how the control laws can be instantiated on a real system.

A. Control Loop

The control loop is depicted in Fig. 2. We first focus on the visual processing pipeline on the right hand side of the dotted line. It starts with capturing a new panoramic image. During each running cycle, the omnidirectional image that is 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 shape.

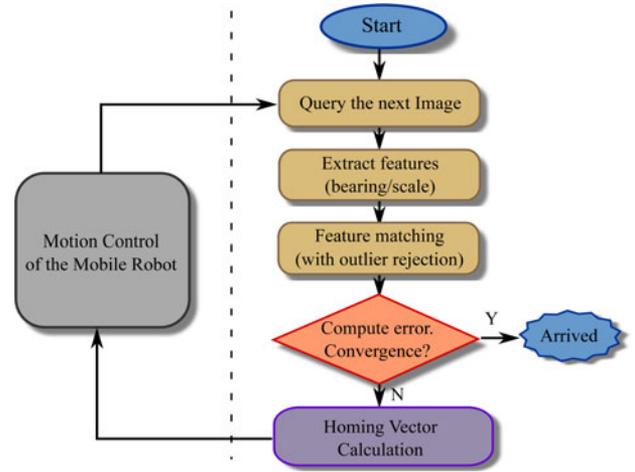


Fig. 2. Control loop of multipoint homing.

SIFT features are then extracted from the unwrapped images and used to compare the current image with the one acquired at the home position. Although there is certainly a distortion in the SIFT features due to mirror nonlinearity, this approach is also used in related works and in particular in [25].

From the matched key points, the visual servoing control law is applied, and a homing vector is computed. This vector is then transmitted to the robot motion controller. The control vector is then implemented while accounting for the nonholonomic constraints of the target robot. The control loop is executed with each newly captured image until convergence.

B. Outlier Rejection

When deploying visual homing algorithms in a real environment, it is likely that the scene captured at the home position will change. For instance, the original image may contain furnitures or objects that are moved over time, e.g., moving people. In general, in an omnidirectional image, these objects only cover a small portion of the image. In this section, we will describe how to remove the falsely matched features, which may be caused by these dynamic objects. Note that if a feature is extracted from the objects, which disappear from the home image, This usually does not raise a significant issue. This is because they usually cannot get matched to anything in the current frame.

The outlier rejection method is the key to enhancing the reliability of our control method against dynamic objects. The underlying assumption is that the main transformation between the current image and the home image is simply a pure rotation. This is true as long as the features are far enough from the camera, or distance between the current position, and the home position is small in comparison to the average distance to the features. In practice, we take every matched feature in the current image, and use the bearing difference with its home correspondence to vote for a potential rotation angle. Once every feature has cast its vote, the resulting histogram contains a main peak corresponding to the main rotation. Other peaks correspond to a set of features that moved coherently between

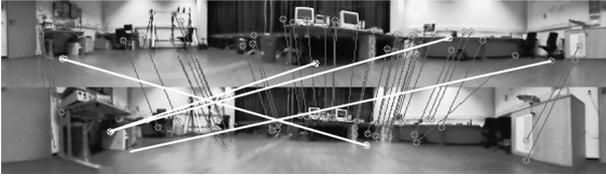


Fig. 3. Result of our outlier rejection algorithm. The gray thin linkages mark the inliers of the matching, and the white linkages mark the outliers.

the two images, i.e., dynamic objects. The background of the histogram refers to outliers or points that violate the hypothesis of a pure rotational error, i.e., points too close to the camera. The assumption may appear a rough approximation of the general case, but used with a lenient threshold, it provides a reliable filter for gross outliers, which would have the worse effect on the control law. In Section VIII-C, we provide an evaluation of the effect of such an assumption. As a result, an instance of outlier rejection using our approach is shown as Fig. 3. The white linkages mark the detected outliers from the raw matching result.

VIII. SIMULATION RESULTS

In this section, we present a number of simulation results highlighting some properties, and basic principles of the homing approaches. The first evaluation is to compare the convergence behavior of different control methods. To this end, it is important to have a common metric to compare all the image-based homing, and characterize the influence of various assumptions and parameters. In particular, we will illustrate the effect of not knowing the distance to the features and the influence of the pure-rotation assumption for outlier rejection.

A. Performance Comparison Using Image-Based Visual Homing

1) *Comparison of Convergence Behavior:* In previous sections, we have discussed four approaches and one related method by [26]:

BOVS: Bearing-only visual servoing, which only takes the bearing angles to the features to compute the control law [see (16)].

SOVS: Scale-only visual servoing, which mainly utilizes the scale information, and the bearing plays a role in the interaction matrix as well [see (15)].

SBVS: Scale and bearing visual servoing uses the full interaction matrix, which integrates the bearing and scale error [see (14)].

HSVS: As a comparison, we also implemented the algorithm in [26], referred to in this paper as HSVS. The implementation follows the summarized equivalent equation:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \sum_i \text{sign}(s_i - s_i^*) \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix}.$$

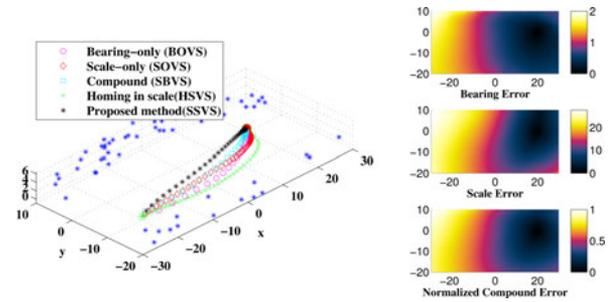


Fig. 4. (Left) Simulated trajectories by different visual homing methods. (Right) Error evaluation on the simulated environment referring to (20,0).

SSVS: The simplified scale-based visual servoing, which also uses scale information but does not require the pseudoinverse computation [see (28)].

Fig. 4 depicts the simulated environment, where visual features (blue stars) are assumed to be located on walls surrounding a ground robot, each with a specified objective scale. The green star marks the start position of the robot, while the red-filled circle is the target home position. A simulated robot uses the observed features to implement the control laws that were presented earlier in the paper, as well as HSVS from [26]. The scale information is computed using a simple pin-hole model, as described in (4). Fig. 4 depicts the incrementally generated trajectories of the five methods.

We tune the controller gain for all the controllers such that the number of iterations required by each method are similar. The four visual servoing trajectories shows that all the control laws guide the robot to the home position in a reasonable manner. However, the proposed SSVS method leads to the straightest trajectory, whereas the HSVS method makes the biggest detour.

The error field of the simulated environment is depicted on the right of Fig. 4, by taking (20,0) as the reference home position. Although the error field of the bearing error appears *flatter* than the scale error field, we could observe that combining bearing and scale error definitely help to shape the error field into a round-shaped potential well.

Fig. 5 provides a graphical view of the normalized error convergence using the various approaches. Note that the first row of the graph represents the bearing error in radian, whereas the other rows represent the error in the scale or combined space. In each graph, the abscissa refers to the number of iterations. As expected, the methods proposed in this study (BOVS, SOVS, SBVS, and SSVS) all show overall exponential convergence, fitting the justification of (2) indicated by [29]. Since the exponential convergence of HSVS is not mathematically guaranteed, the behavior of the error convergence does not reflect such characteristic.

B. Influence of the Assumptions for SSVS

For the visual servoing approaches, the convergence proof relies on the knowledge of the distance to the features. In practice, this assumption cannot be easily fulfilled with a single camera. For the standard visual servoing, it has been shown that

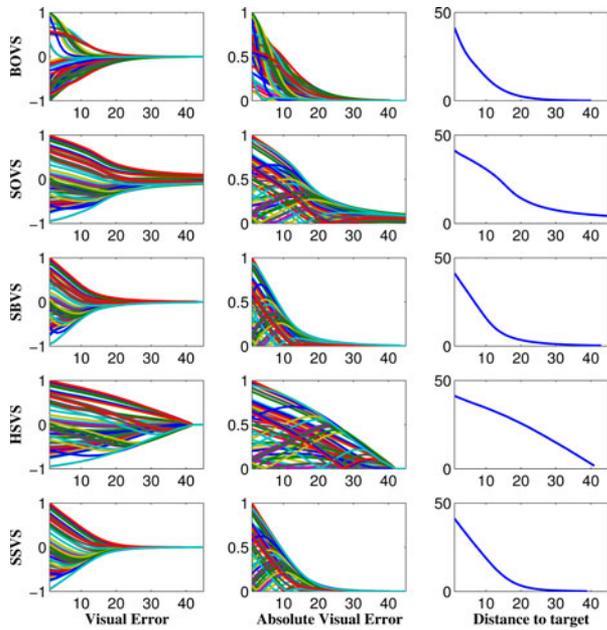


Fig. 5. Visual feature errors and the error distance to the home position over time.

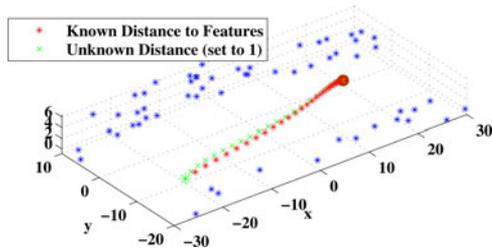


Fig. 6. Comparison between the assumptions with known and unknown distances to features.

assuming that all the features are at unit distance from the camera leads to efficient control strategies [29].

Because we abuse this assumption for SSVS, it is necessary to show its influence on the trajectories. Fig. 6 shows that there is indeed a small difference, but the performance loss, in terms of additional traveled distance, is minor. As for control parameters, different distance assumptions lead to different gains. For the test in Fig. 6, we have tuned the gain so that the number of iterations until convergence is similar in both cases. Taking the same starting and ending positions, by changing the environment features and varying the assumed distance constant, we do the simulation 100 times. We obtain an average traveled distance of 42.1 when the distances are set to unity against 41.3 when knowing the distances. The average absolute curvature are also similar by 0.036 against 0.030. This example shows that similar to the standard visual servoing, the scale-based approach is not sensitive to the known distance assumption, and that distances can be safely approximated to be constant (e.g. 1.0) for real implementations.

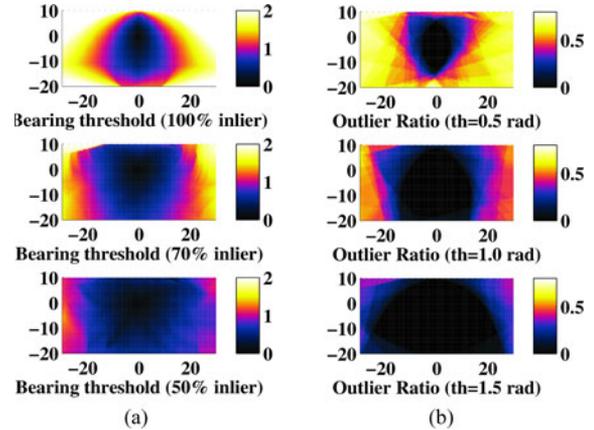


Fig. 7. Justification of the assumption for outlier rejection. (a) Bearing threshold (in rad) for a fixed inlier ratio. (b) Outlier ratio by fixing bearing threshold (in rad).



Fig. 8. Instance at (8,4) from the A1originalH database of [32].

C. Parametrization of the Outlier Rejection

A successful homing process does not depend on a huge numbers of matched features. Therefore, a relatively low inlier ratio, e.g., 50%, is usually more than sufficient to control the robot, given reasonable large numbers of matches. As a reminder, we select inliers if they globally correspond to a pure rotation of the robot, thus ignoring the translational error with respect to the home position. This is done by first finding the best rotation by a voting scheme, and then selecting as inlier the features that correspond to a best rotation approximation, within a given angular threshold.

In order to justify the assumption and evaluate within what region would such an assumption be true, we carry out the test as follows. Considering the simulated environment depicted in Fig. 4, we take the position at coordinate (0,0) as the reference, then evaluate all the other positions (with a resolution of 0.1m) by measuring the numbers of features that do not fit the assumption, namely the outlier ratio.

The analysis result is shown in Fig. 7. The color in Fig. 7(a) implies the minimum required value of the bearing threshold (in radian) for different inlier ratio. This defines the area of the workspace where the transformation between the observed features and the reference can be seen as a pure rotation within a given threshold and for a desired inlier ration (100%, 70%, and 50%). Intuitively, a lower demanded inlier ratio will relax the needs on a precise threshold. Fig. 7(b) depicts an alternative visualization of the results from Fig. 7(a). It shows the ratio of outliers by fixing the rotation threshold. The darker color indicates lower outlier ratio, i.e., potentially better matching results. Taking the lowest figure in Fig. 7(b) as an instance, the dark area implies that by allowing a rotation threshold 1.5 rad, potentially sufficient numbers of matches can be retrieved from

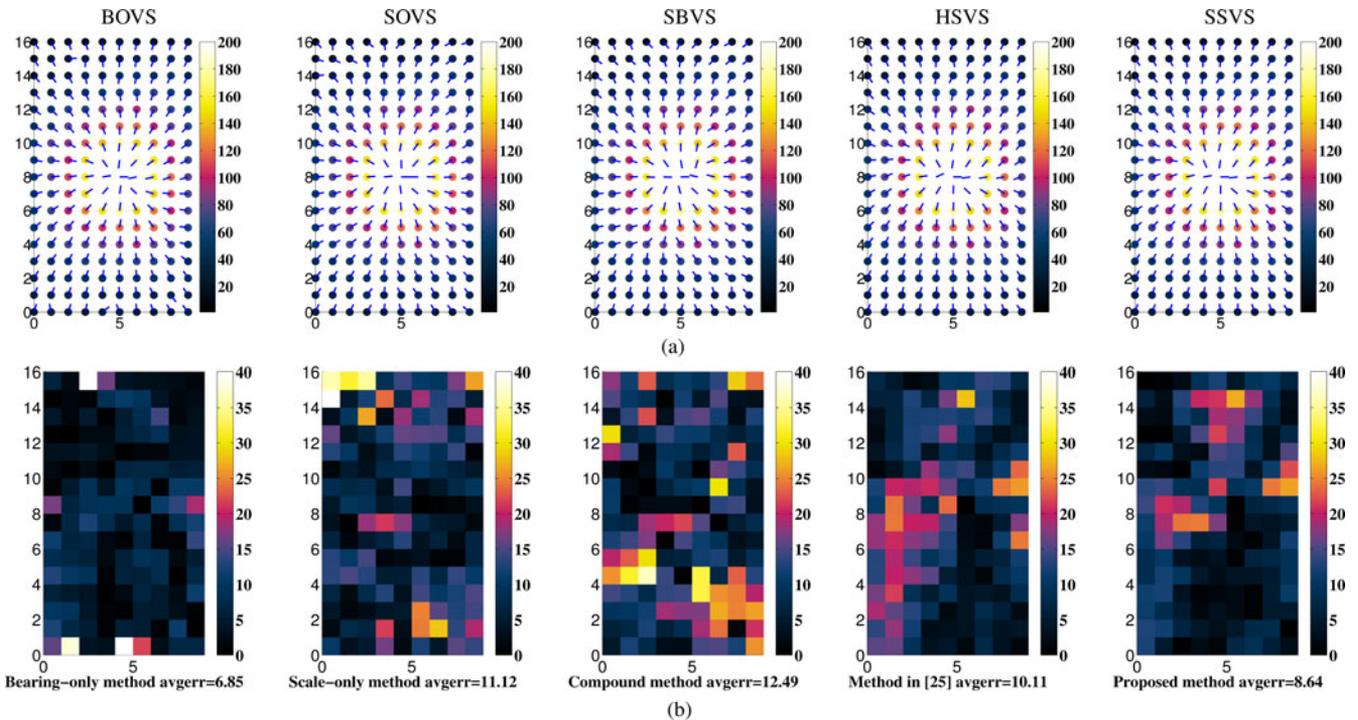


Fig. 9. Homing vectors and error analysis referring to (5.8). The color-map in the first row indicates the number of matched key points; in the second row, the color-map indicates the average angular error in degree. (a) Homing Vectors. (b) AAE analysis (in degree).

a large area around the home position, though it may lead to a high false negative ratio.

For real applications, the rotation threshold needs to be tuned according to the characteristic of the feature distribution in the test environment. Empirically, we choose a threshold of 1.0 rad, resulting in good outlier rejection performance, while still keeping an inlier ratio of more than 80% in the tests on the dataset introduced in Sections IX and X.

IX. RESULTS ON AN INDOOR DATASET

In order to compare all related methods under the same conditions, we test on a widely cited dataset provided by Vardy [32]. The datasets are a collection of panoramic images (some with unwrapped version) obtained from typical indoor environments, plus the calibration information and pose information. Some of the datasets includes dynamic objects in the scene, such as a mobile chair, etc. All the raw images have a resolution of 640×480 (unwrapped image: 561×81) and the actual intervals between the two nearest nodes are equal constants, which is typically 30 cm. An instance from the database is shown in Fig. 8. By taking position (5,8) of A1originalH dataset as the reference home, the homing vectors calculated from other images in the same dataset using different methods are shown as Fig. 9(a). The color of the filled circles indicates the differences in number of matched features. It is interesting to see that SSVS exhibits clean behavior pointing toward the home position, even when the matching ratio is low.

According to the comparison done in [26], the total average angular error (TAAE) can be an important statistic result when evaluating the homing ability. The overall average angular error

(AAE) can be obtained as follows:

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

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

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

The AAE for each position of A1originalH is illustrated in Fig. 9(b).

Test results on the full Vardy datasets are illustrated in Table I, where we show extended statistics of the angular error, such as maximums, minimums, and standard derivations. Darker background on specific numbers mark the best performance of the row. The “**” after the name of datasets indicate that those datasets are unwrapped by detection and operation on the largest circle of the panoramic images. In terms of precision, SSVS and BOVS show the best performance in general. However, we must notice that BOVS requires a good enough estimation of the global heading, which is in general vague using visual compass methods [31]. Moreover, since SSVS and HSVS hold the lowest computational complexity, the advantage of SSVS is revealed.

TABLE I
ERROR ANALYSIS FOR ALL THE ALGORITHMS (IN DEGREE)

Database	BOVS				SOVS				SBVS				HSVS				SSVS			
	TAAE	Max	Min	StdVar																
A1originalH	15.16	50.07	6.52	6.50	14.99	31.90	9.32	3.97	16.04	42.21	10.98	3.43	15.79	30.68	8.97	4.67	12.71	24.98	8.06	3.41
Arboreal*	13.95	35.20	7.06	3.87	20.50	31.63	12.70	4.40	21.26	34.76	12.70	4.13	14.41	24.61	9.63	2.92	12.81	20.44	7.53	2.71
Hall1*	12.97	29.55	6.95	4.67	23.18	35.25	14.28	4.71	24.29	36.97	14.34	5.69	11.73	21.46	8.33	2.03	11.15	19.12	7.55	2.05
Hall2*	21.04	42.03	14.02	4.55	34.28	45.29	25.55	3.91	32.69	41.99	24.83	3.83	17.78	26.01	10.63	3.03	15.24	24.70	9.46	3.08
KitchenH	19.33	40.29	12.17	5.28	26.31	40.60	17.68	5.04	23.63	36.90	14.38	4.58	21.48	38.99	13.34	5.29	20.54	36.94	13.20	4.44
Roeben1H	27.68	48.50	15.17	5.89	25.86	42.11	14.51	5.68	24.74	40.89	13.00	5.71	21.69	42.38	11.01	6.28	19.50	38.56	8.15	6.52
CHall1H	12.21	28.09	6.39	3.96	22.34	37.47	14.20	4.57	22.50	35.85	15.13	4.05	12.26	20.10	8.23	1.94	11.82	18.94	8.35	1.92
CHall2H	18.45	47.63	10.69	5.70	34.16	55.60	24.62	5.43	31.50	49.77	20.83	5.09	19.89	33.63	12.91	3.91	16.63	27.63	9.91	3.05
Chair*	17.66	29.10	8.77	4.15	24.71	41.04	13.08	5.36	23.32	38.03	12.45	4.20	19.28	36.75	11.49	4.28	18.28	35.55	9.53	4.39

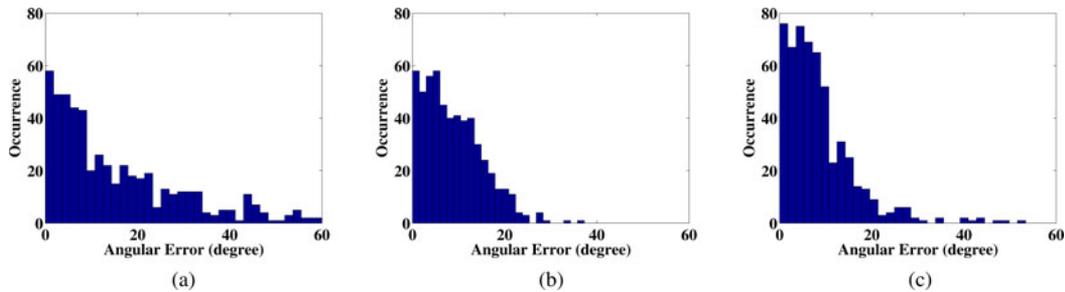


Fig. 10. Histogram of angular errors for the selected algorithms under the test condition with moving people. (a) BOVS. (b) HSVS. (c) SSVS.



Fig. 11. Sample image of the dataset with arbitrarily moving objects.

X. RESULTS OF THE REAL-TIME HOMING TEST

A. Homing Vectors to One Reference

In order to show the performances in a more dynamic environment, a dataset taken together with vicon motion capture system is used for further evaluation. In this test, four people are continuously moving in the scene of which the robot is taking the data. A sample image is shown in Fig. 11.

The calculated homing vectors taking (0,0) position as reference home is depicted in Fig. 12, using SSVS. We could see that the robot trajectory is arbitrary. Thanks to the motion tracking system, the 6 DoF ground truth is simultaneously recorded. It can be noticed that even with relatively low numbers of matched feature points, the robot can show reasonable homing directions using SSVS.

As comparison, only the outperforming algorithms from previous tests—BOVS, SSVS, and HSVS are carried out on this dataset due to the limited space. Concerning BOVS requires global heading estimation, we carry out the evaluation for BOVS using the ground truth heading information. In practice, visual compass is required to provide such information, leading to worse performance. For related works of visual compass, see [31]. The histograms of the angular errors for the three methods are shown in Fig. 10. It indicates that the SSVS has the best precision and BOVS performs the worst even with ground

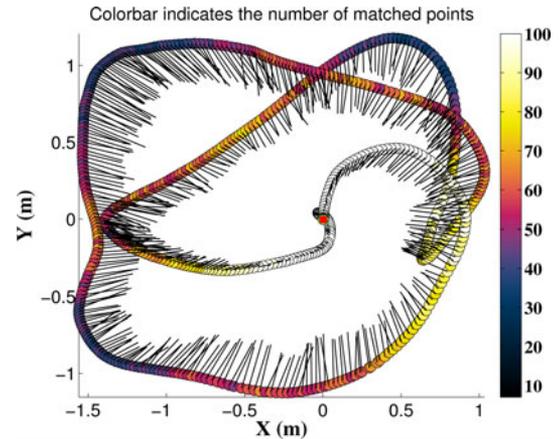


Fig. 12. Results of SSVS with moving people in the scene.

truth heading information. This result shows SSVS is better fit for the robot navigation task in dynamic environments for real-time applications.

B. Experiment of Outlier Rejection

The outlier rejection method is the key to ensuring the correctness of generated homing vectors in dynamic environments. We also compare the cases with and without the proposed outlier rejection method. The statistics of the generated angular errors show the effect of the proposed outlier rejection method, as depicted in Fig. 13. It shows that the implementations with the proposed outlier rejection have lower error mean and smaller derivation. Regarding the simple assumption, the additional computation that is required for outlier rejection is minor. Therefore, such an algorithm is generally feasible for all

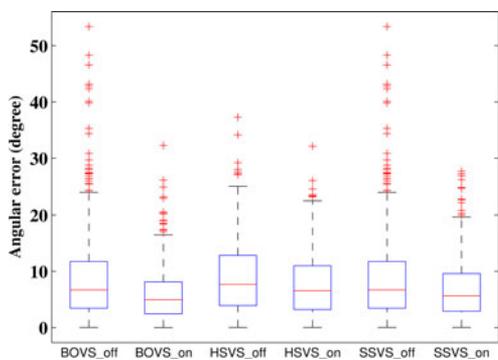


Fig. 13. Error comparison in the cases with/without outlier rejection. The labels marked with `_on` indicate the boxplot with outlier rejection and `_off` indicates the results calculated from the raw matching result.

similar applications using key-point features that are obtained from omnidirectional cameras.

XI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a visual homing framework using visual servoing, which is based on the scale and bearing measurements of popular visual features such as SIFT. After showing how these measurements could be used in the standard visual servoing framework, we proposed a simplified controller with a complexity linear with the number of observed features. We have demonstrated the usability of scale-based visual servoing, and we have shown that our simplified approach is stable and offers better performance than other methods based on the results on standard datasets and experiments in an environment with dynamic objects. We also demonstrated the necessary observability and stability analysis.

Overall, the proposed approach shows certain robustness to dynamic changes in the environment. Similar to any other visual-based homing solution, our approach is sensitive to strong illumination changes, etc., which may completely destroy the appearance. The extended results for a navigation framework using the proposed algorithms are provided in our recent report [33]. Concerning future work, we are considering applying it to outdoor navigation on terrain manifold and extending it to 3-D space for unmanned aerial vehicles.

APPENDIX PUBLIC DATASET

A compact dataset has been captured during the experiment of this study. It is comprised of raw and unwrapped panoramic images captured by an omnidirectional camera. The raw images have a resolution 640×480 , and the unwrapped version has a resolution of 1014×172 . Moreover, the dataset provides pose ground truth with sub-millimeter precision, supported by a vicon motion capture system. A summary is shown in Table II. Please refer to <http://www.asl.ethz.ch/people/lium/personal> for more information.

TABLE II
AVAILABLE PUBLIC DATASET RELATED TO THIS STUDY

Data-set	#Images	Comments	Potential Purpose
CleanRun	876	No dynamic objects	Function test
FourPeople	653	Four moving people	Robustness test
StillCamera	379	Four moving people; the camera keeps position.	Outlier rejection etc.

REFERENCES

- [1] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "Scale-only visual homing from an omnidirectional camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3944–3949.
- [2] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *Int. J. Comput. Vis.*, vol. 33, no. 2, pp. 117–137, 1999.
- [3] M. Franz and H. Mallot, "Biomimetic robot navigation," *Robot. Auton. Syst.*, vol. 30, no. 1–2, pp. 133–154, 2000.
- [4] A. Angeli and J.-A. Meyer, "Incremental vision-based topological SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 22–26.
- [5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. Nat. Conf. Artif. Intell.*, 2002, pp. 593–598.
- [6] S. Thrun. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artif. Intell.* [Online]. 99(1), pp. 21–71. Available: <http://www.sciencedirect.com/science/article/B6TYF-3TOXP9D-2/2/c696e28377beaa9998c9a067cb030cbb>
- [7] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [8] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
- [9] F. Chaumette and E. Malis, "2 1/2 D visual servoing: A possible solution to improve image-based and position-based visual servoings," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 1, pp. 630–635.
- [10] B. Cartwright and T. Collett, "Landmark maps for honeybees," *Biol. Cybern.*, vol. 57, no. 1, pp. 85–93, 1987.
- [11] M. Franz, B. Scholkopf, H. Mallot, and H. Bulthoff, "Where did I take that snapshot? Scene-based homing by image matching," *Biol. Cybern.*, vol. 79, no. 3, pp. 191–202, 1998.
- [12] M. Liu, C. Pradalier, Q. Chen, and R. Siegwart, "A bearing-only 2D/3D-homing method under a visual servoing framework," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 4062–4067.
- [13] P. Corke, "Mobile robot navigation as a planar visual servoing problem," in *Robotics Research: The Tenth International Symposium*. New York, NY, USA: Springer-Verlag, 2003, vol. 6, pp. 361–372.
- [14] D. Lambros, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner, "A mobile robot employing insect strategies for navigation," *Robot. Auton. Syst.*, vol. 30, no. 1, pp. 39–64, 2000.
- [15] T. Goedeme, T. Tuytelaars, L. Van Gool, D. Vanhooydonck, E. Demeester, and M. Nuttin, "Is structure needed for omnidirectional visual homing?" in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2005, pp. 303–308.
- [16] J. Lim and N. Barnes, "Robust visual homing with landmark angles," in *Proceedings of Robotics: Science and Systems*. New York, NY, USA: MIT Press, 2009.
- [17] M. Aranda, G. Lopez-Nicolas, and C. Sagues, "Omnidirectional visual homing using the 1D trifocal tensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2444–2450.
- [18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. vol. 2, Cambridge, U.K.: Cambridge Univ. Press, 2000, ch. 15.
- [19] H. Becerra, G. López-Nicolas, and C. Sagüés, "Omnidirectional visual control of mobile robots based on the 1d trifocal tensor," *Robot. Auton. Syst.*, vol. 58, no. 6, pp. 796–808, 2010.
- [20] H. Becerra, G. Lopez-Nicolas, and C. Sagüés, "A sliding-mode-control law for mobile robots based on epipolar visual servoing from three views," *IEEE Trans. Robot.*, vol. 27, no. 1, pp. 175–183, Feb. 2011.
- [21] G. López-Nicolas, J. Guerrero, and C. Sagüés, "Multiple homographies with omnidirectional vision for robot homing," *Robot. Auton. Syst.*, vol. 58, no. 6, pp. 773–783, 2010.
- [22] A. Cherubini and F. Chaumette, "A redundancy-based approach for obstacle avoidance in mobile robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 5700–5705.

- [23] T. Nierobisch, J. Krettek, U. Khan, and F. Hoffmann, "Optimal large view visual servoing with sets of sift features," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2092–2097.
- [24] F. Hoffmann, T. Nierobisch, T. Seyffarth, and G. Rudolph, "Visual servoing with moments of sift features," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2006, vol. 5, pp. 4262–4267.
- [25] A. Vardy and F. Oppacher, "A scale invariant local image descriptor for visual homing," in *Biomimetic Neural Learning for Intelligent Robots* (Lecture Notes Computer Science). vol. 3575, New York, NY, USA: Springer, 2005, p. 362.
- [26] D. Churchill and A. Vardy, "Homing in scale space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1307–1312.
- [27] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. on Comput. Vis.*, 2006, vol. 3951, pp. 404–417.
- [28] A. Negre, C. Brailion, J. Crowley, and C. Laugier, "Real-time time-to-collision from variation of intrinsic scale," *Algorithm. Found. Robot. VI*, p. 75, 2005.
- [29] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
- [30] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
- [31] A. Vardy, "A simple visual compass with learned pixel weights," in *Proc. IEEE Can. Conf. Electr. Comput. Eng.*, 2008, pp. 001-581–001-586.
- [32] A. Vardy. (2012). "Panoramic image database." [Online]. Available: <http://www.ti.uni-bielefeld.de/html/research/avardy/index.html>
- [33] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "The role of homing in visual topological navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 567–572.



Ming Liu (S'12) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005. During his Master's study at Tongji University, he was a Visiting Scholar with Erlangen-Nürnberg University and Fraunhofer Institute IISB, Germany, for one year. As a result of his performance and under a special grant, in 2007, he was admitted as a Ph.D. student at Tongji University without attaining the master's degree. Since 2009, he has been working toward the Ph.D. degree with the Department of Mechanical Engineering, ETH Zürich, Zurich,

Switzerland.

His current research interests include autonomous mapping, visual navigation, environment modeling, etc.

Mr. Liu received the Best Student Paper Award from the IEEE International Conference on Multisensor Fusion and Information Integration in 2012.



Cédric Pradalier (M'07) received the Ph.D. degree in 2004 from the National Polytechnic Institute of Grenoble, Grenoble, France, on the topic of autonomous navigation of a small urban mobility system. He received the Ingénieur degree from the National Engineering School for Computer Science and Applied Math, Grenoble.

Since November 2007, he has been the Deputy Director with the Autonomous Systems Lab, ETH Zürich, Zurich, Switzerland. He is also the Technical Coordinator of the V-Charge Project (IP, 2010-2014)

and is involved in the development of innovative robotic platforms such as autonomous boats for environment monitoring or prototype space rovers funded by the European Space Agency. He is a founding member of the ETH start-up Skybotix, within which he is responsible for software development and integration. From 2004 to 2007, he was a Research Scientist with The Commonwealth Scientific and Industrial Research Organisation, Highett, Vic., Australia. He was then involved in the development of software for autonomous large industrial robots and an autonomous underwater vehicle for the monitoring of the Great Barrier Reef, Australia.



Roland Siegwart (M'90–SM'03–F'08) received the Diploma degree in mechanical engineering and the Ph.D. degree in mechatronics from the Eidgenössische Technische Hochschule (ETH) Zurich, Zurich, Switzerland, in 1983 and 1989, respectively.

From 1989 to 1990, he was a Postdoctoral Fellow with Stanford University, Stanford, CA, USA. He was then a part time R&D Director with MECOS Traxler AG, Winterthur, Switzerland. He was also a Part-Time Lecturer and the Deputy Head of the Institute of Robotics, ETH Zurich, where, since July 2006, he has been a Full Professor of autonomous systems. In 1996, he was an Associate Professor, and later, a Full Professor for autonomous microsystems and robots with the Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, where he was Co-Initiator and the Founding Chairman of the Space Center EPFL and the Vice Dean of the School of Engineering. In 2005, he held visiting positions with the National Aeronautics and Space Administration (NASA) Ames, Moffett Field, CA, USA, and Stanford University, Stanford, CA. He leads a research group of around 35 people working in the field of robotics, mechatronics, and product design. He is a Coordinator of two large European projects, plays an active role in various rover designs of the European Space Agency, and is a Co-Founder of several spin-off companies.

Dr. Siegwart is a member of the Swiss Academy of Engineering Sciences and a Board Member of the European Network of Robotics. From 2004 to 2005, he was the Vice President for Technical Activities and a Distinguished Lecturer from 2006 to 2007. He was an AdCom Member (2007–2009) of the IEEE Robotics and Automation Society.