

A bearing-only 2D/3D-homing method under a visual servoing framework

Ming Liu, Cédric Pradalier, Qijun Chen and Roland Siegwart

Abstract—Homing is one of the fundamental functions for both the mobile robot and the flying robot. Furthermore, homing can be introduced into a topological navigation system by cyclically setting *Home* positions at the keypoints/nodes in a topological map. In this work, we describe a bearing-only homing method based on only few matching keypoints to grant the mobile robot the homing ability. Our method considers the homing problem as a visual servoing problem in 2D plane and even in 3D space, using an omnidirectional camera as the visual sensor. It doesn't require the distance information to the reference feature points. The proof of the convergence for the algorithm is also given. The simulation results confirm the feasibility and robustness of our method.

I. INTRODUCTION

Robot homing means the robots navigate from a random position to a target *Home* position by sensing only the current visual features and comparing them with the image taken at a home position [9] [24] [1]. It is considered to be one of the basic abilities of a mobile robot, and also one of the most important components of visual navigation in the navigation hierarchy presented by Franz and Mallot [10] especially in topological map based scenarios. The key solution to the robot homing problem is to estimate the "homing vector" which is the direction that robots should follow in order to get to the home position. Practically, the distance to the keypoints/landmarks is not easy to accurately estimate and is costly to measure, e.g. using a heavy and costly laser-range finder. As far as the flying robots are concerned, the payload capability of such robots can not fulfill the requirement of some heavy measurement equipment. A bearing-only homing approach will greatly reduce the dependency of distance measuring devices and the robot can navigate based on only vision in a textured environment. Bearing-only methods enable the robot to navigate by only knowing the angles formed by different landmarks and the robot itself. It requires less computational ability and sensor data from the hardware side as well. This direction is bio-inspired, starting with the work of Cartwright and Colletti [4], namely the so-called 'snapshot' model. Franz *et al.* [11] continued this direction by analyzing the error and convergence properties. Although there are some inevitable

shortcomings, the low computational requirement can still be interesting for certain robots such as flying robots.

As for the autonomous flying robots, it is still an unfledged field. The state-of-art works are mainly focusing on basic mechanisms or control problems, such as attitude control, course stabilisation, and obstacle avoidance [26]. The methods used on flying robots are mainly optical flow [23] and other biomimetics methods [14], which dealt with primary control and tracking problems. Because of the limitation of payload for the flying robots, vision or other lightweight sensors such as IMU are the best choices that can be available. Therefore, the fusion of these sensors seems to be an important method fitting the navigation task. Some recent works, e.g. Steder *et al.* [20], did excellent work in this direction. However, in this paper, we show that pure vision based homing is feasible. We believe that, according to our bounded knowledge, our work is the first that deals with the 3D bearing-only homing problem in 3D space.

"Visual Servoing" [6] has been widely cited in the area of motion control of industrial robots. Our work is stimulated by the work of Corke *et al.* [5], where the author used the ALV[16] (Average Landmark Vector) based method and realized the homing task within the visual servoing framework. ALV method depends on the assumption of the distances to all the landmarks. In our work, we will introduce the bearing-only method under the visual servoing framework, and prove the feasibility in a simulated environment and experiment for both planar and 3D cases.

We will stress the following contributions of our work:

- 1) The proof of convergence for the bearing-only homing;
- 2) The first work on the 3D bearing-only homing;
- 3) A bearing-only homing method, which enables the homing task in a featured environment;
- 4) Combine the homing task for mobile robots with the pure IBVS (Image Based Visual Servoing), without the structure reconstruction, by a low cost method.

In the following, we first give an overview of related work in Section II, and then, we define the problem of homing under the IBVS (Image-based Visual Servoing) frame in Section III. Sections IV and V describe the algorithm and control strategy of our approach, together with the simulation results. Section VI will introduce the experimental results. The discussion of the results is shown in the Section VI.

II. RELATED WORK

The latest result in navigation for autonomous flying robot is given by Grzonka *et al.* [13]. They proposed a complete quadrotor system using laser sensor fusing with an IMU.

This work was supported by CSC (China Scholarship Council) and Robots@home STREP Project IST-6-045350, and partially supported by The National high technology Research and Development Program of China (863 Program) 2009AA04Z213

Ming Liu is with Autonomous Systems Laboratory in ETH Zurich and CEIE of Tongji University ming.liu@mavt.ethz.ch

Cédric Pradalier and Roland Siegwart are with Autonomous Systems Laboratory in ETH Zurich, Qijun Chen is with CEIE of Tongji University.

Based on this system, they managed to solve the SLAM problem in real time. Our method will be based on only vision sensors. Comparing to the work done by Bekris *et al.* [2], our method doesn't depend on the certain requirement of "three landmarks should be detected and corresponded at different position"; this method works by randomly select features from the dynamic real-world using an incremental concept. It enables the homing ability without the tracking of fixed features. In this paper we also will mathematically prove the convergence of the bearing-only method.

Considering the homing vector extraction, one of the famous and widely cited methods is the landmark-based ALV proposed by Lambrinos *et al.* [16]. It converts the homing problem to a vector operation process, by summing up the vectors to a number of keypoints and calculating the error of the reference vectors of current position and home position. But it partly depends on the geometric knowledge of the landmarks and environment, which makes it dependent to the distances to the keypoints. Zhang *et al.* [25] dealt with the automatic calibration and navigation by visual servoing on geometric features such as vanishing points and line orientations. It can be considered as an application of position based visual servoing in robot navigation. Kirigin *et al.* [15] used two horizontally set fisheye cameras for the homing task in an outdoor environment. They showed that the more reference points that are used, the more robust and better trajectory the robot can get. We will extend our method in this direction in the conclusion. Goedeme *et al.* [12] proposed that the structure of the environment is not required in the robot homing, and they estimated the ratio of the distances to the matching keypoints by triangulation. In this work, we mathematically propose that even the distance to the keypoint can be neglected and a bearing-only homing method is feasible to implement. Usher *et al.* [22][21] continued Corke's theoretical results and extended their work on a car-like vehicle. As the estimated distance to the landmarks are essential for the homing method, they introduced the distance information directly to the ALV algorithm and got pretty satisfying results. Our work will show the special competitive strength in the situation when the distance to the landmarks are hard to estimate. By extracting and tracking the features with normalized RGB color model, Fitzgibbons *et al.* [8] dig into the bearing-only SLAM problem for an outdoor environment. It proved that the bearing-only approach can provide a robust solution to the SLAM problem. The latest result is done by Lim *et al.* [17]. They divided the 2D planar surface into four regions and estimated the current robot position by measuring the bearings of landmarks. The theory was proved by sketch drawing. We will give a mathematical proof to our method.

III. PROBLEM DEFINITION

A. Formulation

Let's start with the 2D planar case. The problem is defined as Fig. 1¹, where P_1, P_2, P_3 are three keypoints which can be

detected from current position O and home position H. The purpose is to guide the robot from O to H only by comparing the separation angles β_i and target angles β_i^* , $i = 1, 2, 3$. β_1 is the angle formed by $\overrightarrow{OP_1}$ and $\overrightarrow{OP_3}$ and β_2 is the angle of vectors $\overrightarrow{OP_1}$ and $\overrightarrow{OP_2}$ etc.

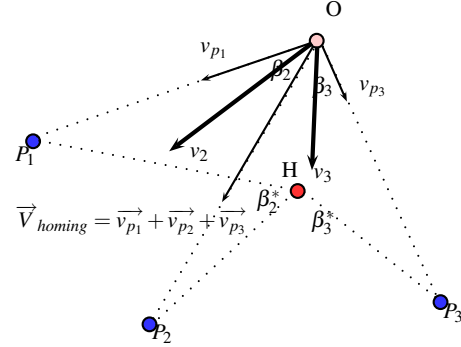


Fig. 1. Problem Definition

The operation of the robot can be summarized as the control of its velocity vector. Particularly, we choose the bisector of the angle β_1 as the direction to control the angle β_1 , which we set as v_1 . Meanwhile, v_2 is the controlled velocity along the bisector of β_2 , v_3 for β_3 . We need to find the proper J to fit $\dot{\epsilon} = Ju$, where ϵ is the error of observed and desired angles, as shown in Eq. 1. In this application, the controller should fit:

$$\begin{pmatrix} \beta_1 - \beta_1^* \\ \beta_2 - \beta_2^* \\ \beta_3 - \beta_3^* \end{pmatrix} = Ju = J \begin{pmatrix} v_{p1} \\ v_{p2} \\ v_{p3} \end{pmatrix} \quad (1)$$

In 3D space, a minor change of the formulation is that at least four feature points are needed, because each pair of unrelated features can provide only one standalone equation. Assuming that there is a fourth feature P_4 in the 3D space, and the angle formed by $\overrightarrow{OP_1}$ and $\overrightarrow{OP_4}$ is β_4 , and the angle formed by $\overrightarrow{HP_1}$ and $\overrightarrow{HP_4}$ is β_4^* . v_4 is the needed speed along the bisector of angle $\angle P_1OP_4$, as shown in Fig. 2.² Therefore, the relationship between the velocity and feature error is shown as below:

$$\begin{pmatrix} \beta_1 - \beta_1^* \\ \beta_2 - \beta_2^* \\ \beta_3 - \beta_3^* \\ \beta_4 - \beta_4^* \end{pmatrix} = Ju = J \begin{pmatrix} v_{p1} \\ v_{p2} \\ v_{p3} \\ v_{p4} \end{pmatrix} \quad (2)$$

As shown in Fig. 1, the homing velocity,

$$\vec{V}_{homing} = \vec{v}_p + \vec{v}_q + \vec{v}_m \quad (3)$$

B. Why can the target position be defined only by angles

Let's review Fig. 1. Both β_2^* and β_3^* can be considered as angle of arc $\widehat{P_1P_3}$ in $\odot P_1P_3H$ and $\widehat{P_3P_2}$ in $\odot P_3P_2H$ respectively. According to the Equal angle, equal arc theorem, if points $P_1P_2P_3H$ are concyclic, there will

¹ v_3 is omitted in the sketch because of the limited space

²The bisector vectors $v_i, i = 1, 2, 3, 4$ are omitted in Fig. 2

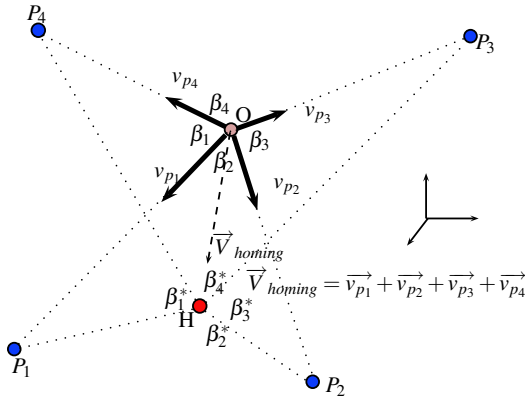


Fig. 2. The model of 3D bearing-only homing

be infinite solution to this problem, which is physically impossible; on the other hand, if not, as shown in Fig. 3

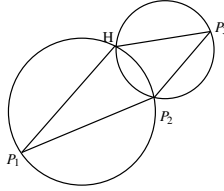


Fig. 3. Intersection of two circles

existing and only existing two intersection points of two intersectant circles, i.e. H and P_2 . Therefore, H can be well-determined in a 2-D planar. In this case, O and H are on the same relative side of points $P_1 P_2 P_3$, so that if the angles β_2 or β_3 is smaller (or greater) than the desired value, all we need to do is simply control the angle to the desired value, without thinking of the symmetric problem. In the 3D case, instead of two circles, there are three spheres intersecting each other. The intersections will be either a circle (when the three spheres are coaxial, which means at least three are coaxial with the current robot position. In that case, only three features can be observed, conflicting with our requirement for minimum number of features.) or two intersection points. One of the two points is our target position, and the other is coincident with one of the features. Therefore, with two (planar case) or three (3D space case) or more bearings, the position of homing position can be well-determined.

IV. CONVERGENCE PROOF

Before we start solving this control matrix J , let's prove this method can converge. Considering the situation shown in Fig. 4. It shows the variation of separation angle θ during the moving of robot from G to G' , $\theta \in (0, 2\pi)$. The distance of these two points $d = vdt$, where v is the speed of translation and dt is the time diffusion. Therefore, the angle variation

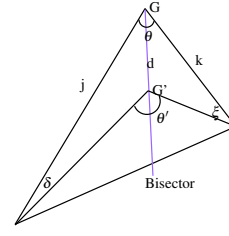


Fig. 4. The changing of separation angle

$\Delta\theta = \theta' - \theta = \delta + \xi$. According to the sine theorem,

$$\begin{cases} \frac{d}{\sin \delta} = \frac{j}{\sin(\pi - \delta - \frac{1}{2}\theta)} \\ \frac{d}{\sin \xi} = \frac{k}{\sin(\pi - \xi - \frac{1}{2}\theta)} \end{cases} \quad (4)$$

Because $t \rightarrow 0 \implies d \rightarrow 0$, and $\delta, \xi \rightarrow 0$, the equation above can be simplified as,

$$\begin{cases} \frac{d}{\delta} = \frac{j}{\sin \frac{\theta}{2}} \\ \frac{d}{\xi} = \frac{k}{\sin \frac{\theta}{2}} \end{cases} \quad (5)$$

and the error of measurements

$$\varepsilon = |\theta' - \theta^*| - |\theta - \theta^*| \quad (6)$$

if $\theta > \theta^*$,

$$\varepsilon = d \sin \frac{\theta}{2} \left(\frac{1}{j} + \frac{1}{k} \right)$$

if $\theta < \theta^*$,

$$\varepsilon = -d \sin \frac{\theta}{2} \left(\frac{1}{j} + \frac{1}{k} \right).$$

It is without lose of generality in setting $j = k = 1$, since we have no idea of the distance in practical. The derivative of the angle error is

$$\dot{\varepsilon} = \begin{cases} 2v \sin \frac{\theta}{2}, & \theta > \theta^* \\ -2v \sin \frac{\theta}{2}, & \theta < \theta^* \end{cases} \quad (7)$$

To make sure $\dot{\varepsilon} < 0$, v should be positive if $\theta > \theta^*$, vice versa. The negative v means that the robot should move along the inversed direction of the bisector, to decrease the separation angle. Since $\frac{\theta}{2} \in (0, \frac{\pi}{2})$, by setting the correct moving direction, it can guarantee that $\varepsilon \rightarrow 0$. It shows that the velocity control in the bisector direction (or inversed direction) can enable the angle to converge to any demanded target. In 3D case, the problem is similar except that the separation angle should be instead by the solid angle of polygon formed by the feature points; the distance to the referred segment is instead by the area of the polygon. According to Section III, the bearing angles can determine the positions of robots in both 2D and 3D space, therefore the bearing-only homing method can work in both cases.³

³In 3D case, the problem is almost the same, by changing the bearing angles to solid angles in 3D. The proof process is omitted here.

V. CONTROL STRATEGY AND SIMULATION

In section IV, we have proved that if the robot tries to move along the bisector of separation angle, we can reach a stable desired bearing. The question now is how to represent and calculate the velocity vector to lead the robot to achieve the target, especially the homing direction. Additionally, how to prove the stability of a controller using this kind of bearing-only method.

A. Velocity Decomposing and Controller Convergence

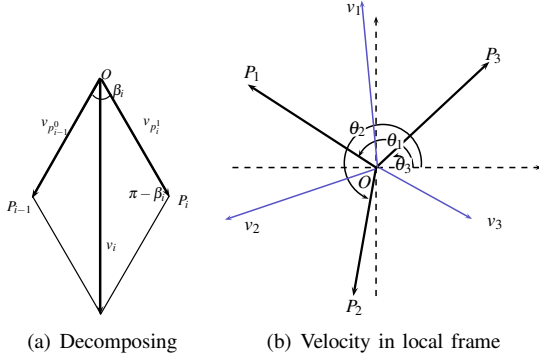


Fig. 5. Velocity Decomposing

We bring the velocity components into a local frame at the robot. The sketch is shown in Fig. 5. θ_i represents the separation angle of $(0,1)$ to $\overrightarrow{OP_i}$. As the speed along the bisector of $\angle P_{i-1}OP_i$ can be equally decomposed along the direction of OP_{i-1} and OP_i , the speed along the bisector,

$$v_i = 2(\beta_i - \beta_i^*) \cos \frac{\beta_i}{2}. \quad (8)$$

where $\beta_{ij} = \theta_i - \theta_j$, and the parameters with * are the observation from the *Home* position. Therefore, the components:

$$v_{p_{i-1}}^0 = v_{p_i}^1 = \beta_i - \beta_i^* \quad (9)$$

According to the decomposing map in Fig. 5(b), the speed component along OP_i direction can be represented as,

$$\overrightarrow{v_{p_i}} = \overrightarrow{v_{p_i}^0} + \overrightarrow{v_{p_i}^1} \quad (10)$$

If we represent the output of our controller by the components in the directions of all the bisectors, it is shown in Fig. 6⁴, where $\theta_0 = \theta_3$, and $\theta_4 = \theta_1$. Hereby we set up two lemma for the proof of convergency for our controller.

Lemma 1: According to the proof in the previous section, when the robot moves along the bisector of angle $\angle P_1OP_2$, it is always possible to choose a velocity v , such that $\varepsilon \rightarrow 0$.

Lemma 2: Lemma 1 extends to 3D by moving on the median of the tetrahedron $OP_1P_2P_3$, and choosing a velocity such that the solid angle β defined by the tetrahedron converges to β^* .

The proof of the convergency is separated in 2D and 3D:

- **In 2D:** In the 2D case, let's assume the case with point P_1, P_2 and P_3 , bisector $\overrightarrow{v_1}, \overrightarrow{v_2}$ and $\overrightarrow{v_3}$. If the points P_i are distinct, $\overrightarrow{v_1}, \overrightarrow{v_2}$ becomes a base of the plane. Any

⁴ $\overrightarrow{OP_i}$ is the unit vector from O towards P_i

velocity V in the plane can be expressed in this base as $\overrightarrow{V} = a\overrightarrow{v_1} + b\overrightarrow{v_2}$. When implementing a control law to control $\varepsilon_1, \varepsilon_2$ and ε_3 to zero, using Lemma 1, we can freely select a and b such that ε_1 and ε_2 converge to zero. We now just have to show that ε_3 converges to zero. This comes from the fact that $\beta_1 + \beta_2 + \beta_3 = \pi$ and $\beta_1^* + \beta_2^* + \beta_3^* = \pi$ as well. So, when $\varepsilon_1 \rightarrow 0$ and $\varepsilon_2 \rightarrow 0$, then $\beta_1 \rightarrow \beta_1^*$ and $\beta_2 \rightarrow \beta_2^*$. From these results, we can refer that $\beta_3 \rightarrow \beta_3^*$.

- **In 3D:** In the 3D, with $\beta_i; i = 1 \dots 4$ the four solid angle defined by $GP_iP_{i+1}P_{i+2}$, if $\overrightarrow{v_i}$ is a vector on the median of the $GP_iP_{i+1}P_{i+2}$ tetrahedron (equiv. to the bisector in 2D), and the points are distinct then $\overrightarrow{v_i}; i = 1 \dots 3$ is a base of the volume. Using Lemma 2, we can always find a velocity $\overrightarrow{V} = \sum_{i=1}^3 a_i \overrightarrow{v_i}$ such that $\varepsilon_i; i = 1 \dots 3 \rightarrow 0$. So we now have to show that $\varepsilon_4 \rightarrow 0$ as well. This comes from the fact that at G , we have $\sum_{i=1}^4 \beta_i = 4\pi$ steradians, and at G^* , we have $\sum_{i=1}^4 \beta_i^* = 4\pi$ as well. As in 2D, the convergence of β_i to β_i^* for $i = 1 \dots 3$ leads to the convergence of β_4 to β_4^* .

Theoretically, if the controller is convergent, the error norm should converge to 0, i.e.

$$\|\varepsilon_i\|^2 \rightarrow 0 \quad (12)$$

namely,

$$\dot{\varepsilon}_i \cdot \varepsilon_i < 0 \quad (13)$$

Because v_1, v_2, v_3 are all along the bisector directions, and v_i is set to the direction to eliminate the error of observed angle as shown in Eq. 8, these components of the output $\overrightarrow{V}_{homing}$ are fit the applicable sphere of Eq. 7. Combining Eq. 11 with Eq. 7 and Eq. 13, the derivative of the angle errors are:

$$\dot{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} = -4 \begin{pmatrix} \cos \frac{\beta_{13}}{2} \sin \frac{\beta_{13}}{2} (\beta_{13} - \beta_{13}^*) \\ \cos \frac{\beta_{21}}{2} \sin \frac{\beta_{21}}{2} (\beta_{21} - \beta_{21}^*) \\ \cos \frac{\beta_{32}}{2} \sin \frac{\beta_{32}}{2} (\beta_{32} - \beta_{32}^*) \end{pmatrix} \quad (14)$$

So that the derivative of the norm of error

$$\dot{\varepsilon} \cdot \varepsilon = -2 \begin{pmatrix} \sin \beta_{13} \cdot (\beta_{13} - \beta_{13}^*)^2 \\ \sin \beta_{21} \cdot (\beta_{21} - \beta_{21}^*)^2 \\ \sin \beta_{32} \cdot (\beta_{32} - \beta_{32}^*)^2 \end{pmatrix} \quad (15)$$

when the measured separation angle $\beta_{ij} \in (0, \pi)$, it fits Eq. 13, therefore all the components of the controlled speed output converge. Based on the linear relation of $\overrightarrow{V}_{homing}$ and v_i , the control method is convergent too.

B. Velocity Definition

One common idea is that when the robot approaches the target, it should move much more slowly to decrease the error caused by mechanism. Without lose of generality, we assume that the angle error and the speed to control the angle error along the bisectors are with the relationship shown in Eq. 8. The speed along each bisector can be decomposed into the directions to the feature points. The relation of the

$$\vec{V}_{homing} = \sum_{i=1}^3 \vec{v}_i, \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vec{v}_3 \end{pmatrix} = 2 \begin{pmatrix} \cos \frac{\beta_{13}}{2} (\beta_{13} - \beta_{13}^*) & 0 & \cos \frac{\beta_{13}}{2} (\beta_{13} - \beta_{13}^*) \\ \cos \frac{\beta_{21}}{2} (\beta_{21} - \beta_{21}^*) & \cos \frac{\beta_{21}}{2} (\beta_{21} - \beta_{21}^*) & 0 \\ 0 & \cos \frac{\beta_{32}}{2} (\beta_{32} - \beta_{32}^*) & \cos \frac{\beta_{32}}{2} (\beta_{32} - \beta_{32}^*) \end{pmatrix} \begin{pmatrix} \vec{OP}_1 \\ \vec{OP}_2 \\ \vec{OP}_3 \end{pmatrix} \quad (11)$$

Fig. 6. Output Velocity

linear speed towards feature points P_1, P_2, P_3 and the feature errors $\beta_{ij} - \beta_{ij}^*$ can be written as:

$$\begin{pmatrix} \vec{v}_{p_1} \\ \vec{v}_{p_2} \\ \vec{v}_{p_3} \end{pmatrix} = \begin{pmatrix} \vec{OP}_1 & \vec{OP}_1 & 0 \\ 0 & \vec{OP}_2 & \vec{OP}_2 \\ \vec{OP}_3 & 0 & \vec{OP}_3 \end{pmatrix} \begin{pmatrix} \beta_{13} - \beta_{13}^* \\ \beta_{21} - \beta_{21}^* \\ \beta_{32} - \beta_{32}^* \end{pmatrix} \quad (16)$$

The output of the controller can be formulated as,

$$\vec{V}_{homing} = \sum_{i=1}^3 \vec{v}_{p_i}. \quad (17)$$

Comparing to the classic Visual Servoing method, we use Eq.17 directly in a feedback control loop. The simulation result is shown in Fig. 7.

All the feature points (in red), *home* position and current position are random generated in the simulated environment. The blue points are the potential feature points candidates. In the beginning of every running cycle, a new group of feature points will be selected, to generate the homing vector. It simulates the situation in real application, when the robot can not easily track all the feature points. Actually, it's not necessary for the robot to track the features with such an incremental method. With the same manner, the 3D homing

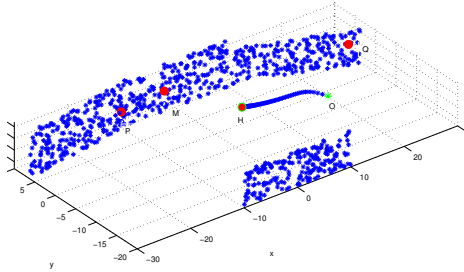


Fig. 7. The result of 2D bearing-only homing in simulated environment. Considering the huge amount of keypoints that we can get in a practical application, we build a 4-wall environment accordingly. The algorithm selects randomly three keypoints out of 800 at the beginning of each running cycle. The average error of the final position is smaller than 1 distant unit.

equation can be calculated by summing up the velocities along the bisectors of bearings. The relation equation is shown in Fig. 18. The parameter definition is the same as Fig. 2. The output of the controller is shown in Eq. 19.

$$\begin{pmatrix} v_{p_1} \\ v_{p_2} \\ v_{p_3} \\ v_{p_4} \end{pmatrix} = -2 \begin{pmatrix} \cos \frac{\beta_1}{2} & 0 & 0 & \cos \frac{\beta_4}{2} \\ \cos \frac{\beta_1}{2} & \cos \frac{\beta_2}{2} & 0 & 0 \\ 0 & \cos \frac{\beta_2}{2} & \cos \frac{\beta_3}{2} & 0 \\ 0 & 0 & \cos \frac{\beta_3}{2} & \cos \frac{\beta_4}{2} \end{pmatrix} \begin{pmatrix} \Delta\beta_1 \\ \Delta\beta_2 \\ \Delta\beta_3 \\ \Delta\beta_4 \end{pmatrix} \quad (18)$$

$$\vec{V}_{homing} = \sum_{i=1}^4 v_{p_i} \vec{OP}_i \quad (19)$$

The simulation results is shown in Fig. 8(a) and Fig.8(b).

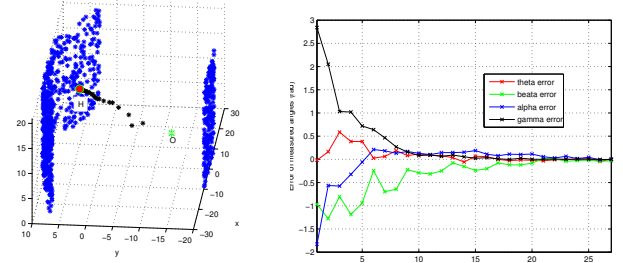


Fig. 8. The simulation and errors of the four bearing angles on Eq. 19

To improve the method, as shown in Eq. 18, all the multipliers on the right side are unrelated. Therefore, in practice, more feature points can be added following the same structure to enhance the robustness of the algorithm.

VI. EXPERIMENT

The experiment was carried out in a lab environment using a 150 degree fisheye camera. The algorithm is realized in C with the OpenCV library [3]. We only show the result of our method in 3D space, because it can basically stand for the real application and the 3D case is more complicated than planar. The result is shown in Fig. 9.

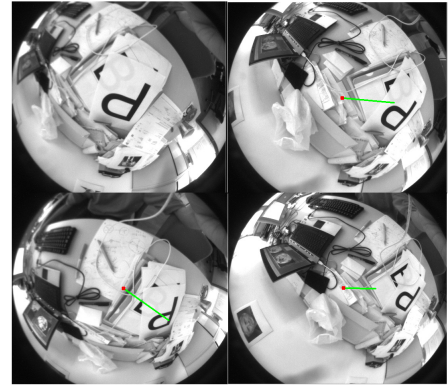


Fig. 9. The experiment result for the 3D case. The top-left image is the image take at the "Home" position, and the rest are taken at places nearby. The red dot in the center marks the estimated position of the flying robot. The green lines mark the homing direction, which is a projection of the real 3D homing direction on the image planar.

In the experiment, all the matched features are included in generating the homing direction as mentioned in the previous section. It expanded the matrix shown in Eq. 18 to N -dimension, where N is the number of observed keypoints. There are some failure cases, which are caused by observation noise and outliers, are not shown in the Fig.

9. We believe the failure caused by outliers can be eliminated efficiently by RANSAC [7] or other techniques. A detailed evaluation will be given in our further research.

VII. DISCUSSION AND CONCLUSION

The 2D results in Fig. 7 show that the bearing-only homing method based on 3 features (keypoints) is feasible for the homing task for a mobile robot. Meanwhile, the results in Fig. 8(a) and Fig. 8(b) show that both the methods in Eq. 18 can manage to finish the homing task in 3D space for a flying robot. More than 100 times of simulation have been taken with random starting position and random homing position in the 3D space. The simulated trajectories can all arrive the home position within 50 iterations, with a final position error within 1 distance unit. The error records shown in Fig. 8(b) states that the method shows a convergence in eliminating all the 4 separation angles. The experiment result shows that our method is applicable in 3D homing task, and further evaluations are still needed.

Because this method depends on the bearings of features without the requirement of very specified keypoint descriptors. Several existing fast descriptors can also be used, especially in the 2D case, such as the vertical line based method by Scaramuzza *et al.* [19] and adaptive color based descriptors by Liu *et al.* [18]

In this paper, two methods are described to be effective in solving the bearing-only homing problem for both mobile robots and flying robots. We employed the IBVS problem as a reference, and developed our bearing-only solution in both 2D and 3D. These techniques enable the homing capability for the robots, without any distance information. The simplified transition matrix will help to form an easily structured closed-loop control system, with flexible expandability.

As for the problem of 3D homing, we should confess that more topics should be discussed, e.g. the control problem for the flying robots is not as accurate or robust as the control for mobile robots, and the pose estimation is also very important for a bearing-only method. Nevertheless, this work originally states the bearing-only homing problem for the flying robot in 3D space, and proposed a robust homing algorithm.

REFERENCES

- [1] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2):117–137, 1999.
- [2] K.E. Bekris, A.A. Argyros, and L.E. Kavraki. Exploiting Panoramic Vision for Bearing-Only Robot Homing. *Computational Imaging and Vision*, 33:229, 2006.
- [3] G. Bradski. The openCV library. *DOCTOR DOBBS JOURNAL*, 25(11):120–126, 2000.
- [4] B.A. Cartwright and T.S. Collett. Landmark maps for honeybees. *Biological Cybernetics*, 57(1):85–93, 1987.
- [5] P. Corke. Mobile robot navigation as a planar visual servoing problem. In *Robotics Research: The Tenth International Symposium*, volume 6, pages 361–372. Springer, 2001.
- [6] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
- [7] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 1981.
- [8] T. Fitzgibbons and E. Nebot. Bearing only SLAM using colour-based feature tracking. In *Proceedings of the Australasian Conference on Robotics and Automation, Auckland*, 2002.
- [9] N. Franceschini, JM Pichon, C. Blanes, and JM Brady. From insect vision to robot vision [and discussion]. *Philosophical Transactions: Biological Sciences*, pages 283–294, 1992.
- [10] M.O. Franz and H.A. Mallot. Biomimetic robot navigation. *Robotics and autonomous Systems*, 2000.
- [11] M.O. Franz, B. Scholkopf, H.A. Mallot, and H.H. Bulthoff. Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, 79(3):191–202, 1998.
- [12] T. Goedeme, T. Tuytelaars, L. Van Gool, D. Vanhooydonck, E. De-meester, and M. Nuttin. Is structure needed for omnidirectional visual homing? In *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pages 303–308, 2005.
- [13] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.
- [14] M. Ichikawa, H. Yamada, and J. Takeuchi. Vision and Mechatronics. Flying Robot with Biologically Inspired Vision. *J Rob Mechatron*, 13(6):621–624, 2001.
- [15] I. Kirigin and S. Singh. Bearings based robot homing with robust landmark matching and limited horizon view. Technical report, Technical Report CMU-RI-TR-05-02, Carnegie Mellon University, 2005.
- [16] D. Lambrinos, R. Moller, R. Pfeifer, and R. Wehner. Landmark navigation without snapshots: the average landmark vector model. In *26th Goettingen Neurobiology Conference*, volume 1, page 221, 1998.
- [17] J. Lim and N. Barnes. Robust visual homing with landmark angles. In *Proceedings of Robotics: Science and Systems*, 2009.
- [18] Ming Liu, Roland Siegwart Davide Scaramuzza, Cedric Pradalier, and Qijun Chen. Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2009.
- [19] D. Scaramuzza, N. Criblez, A. Martinelli, and R. Siegwart. Robust feature extraction and matching for omnidirectional images. In *Field and Service Robotics: Results of the 6th International Conference (STAR: Springer Tracts in Advanced Robotics Series Volume 42)*, volume 42, pages 71–81. Springer, 2008.
- [20] B. Steder, G. Grisetti, C. Stachniss, and W. Burgard. Visual SLAM for Flying Vehicles. *IEEE Transactions on Robotics*, 24(5):1088–1093, 2008.
- [21] K. Usher, P. Corke, and P. Ridley. Home alone: Mobile robot visual servoing. In *WS2: Workshop on Visual Servoing*, page 10, 2002.
- [22] K. Usher, P. Ridley, and P. Corke. Visual servoing of a car-like vehicle—an application of omnidirectional vision. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, volume 3, 2003.
- [23] A. Vardy and R. Moller. Biologically plausible visual homing methods based on optical flow techniques. *Connection Science*, 17(1):47–89, 2005.
- [24] K. Weber, S. Venkatesh, and M. Srinivasan. Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65, 1999.
- [25] Z. Zhang, R. Weiss, and A.R. Hanson. Automatic calibration and visual servoing for a robot navigation system. In *Proceedings of IEEE International Conference on Robotics and Automation, 1993.*, pages 14–19 vol.1, May 1993.
- [26] J.C. Zufferey. *Bio-inspired vision-based flying robots*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2005.