BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 2, 2019
DOI: 10.24425/bpas.2019.128114

Mobile robot visual homing by vector pre-assigned mechanism

X. JI*, Q. ZHU, J. WANG, C. CAI, and J. MA

College of Automation, Harbin Engineering University, 150001, China

Abstract. In this paper, we present an optimization mechanism for two popular landmark-based mobile robot visual homing algorithms (ALV and HiSS), called vector pre-assigned mechanism (VPM). VPM contains two branches, both of which can promote the homing performance effectively. In addition, to make the landmark distribution satisfy the equal distance assumption, a landmark optimization strategy is proposed based on imaging principle of the panoramic vision. Experiments on both panoramic image database and a real mobile robot have confirmed the effectiveness of the proposed methods.

Key words: robot navigation, visual homing, panoramic vision, ALV, HiSS.

1. Introduction

The topic of autonomous mobile robot navigation has been widely studied and discussed for decades in robotic community. Researchers adopt different types of sensors (such as gyroscopes, laser radars, cameras, etc.) to extract valuable information, guiding the robot to the specified position [1–3]. Among the numerous vision-based robot navigation approaches, visual homing has received widespread attention for its simple model and high navigation precision. So far, a variety of visual homing approaches has been proposed and proved to exhibit great homing performance [4, 5].

Visual homing is a qualitative navigation technology. Compared to traditional SLAM (simultaneous localization and mapping) [6–8] techniques, visual homing does not require the robot to perform any self-localization or mapping, but only to compare the two panoramic images respectively captured at its current location and the reference home location, thus greatly simplifying the sources of both the hardware and the software [9–11]. Home vector is the only output of the visual homing approaches, which refers to the direction from the robot's current location to the home location. Therefore, visual homing is a noteworthy robot navigation technology that is suitable for practical applications.

With the development of technology, many advanced visual homing algorithms were presented by combining other related equipment or knowledge, such as odometer information [5], depth sensors [12], machine learning [13], etc. According to the different input forms, visual homing can be broadly divided into two categories, including pixel-based homing and landmark-based homing.

Pixel-based homing takes the pixels of the specific positions in the images as input. The most representative pixel-based homing approach is warping. The warping methods distort the goal image based on three parameters, which describe the direction, rotation and distance of the robot's movement from the home location to the current location. All the distorted images are compared with the current image, and the optimal parameter combination can be obtained when the differences (i.e. Euclidean distance) between the two images are minimal. The home vector is then calculated by the above parameter combination [14–17].

Landmark-based homing takes the distinctive features by feature extraction and matching algorithms (such as SIFT [18] and SURF [19]) as input. The most two representative landmark-based homing approaches are ALV (average landmark vector) and HiSS (homing in scale space). ALV assumes that the robot is no longer storing the entire scene it observes, but the landmark vectors pointing from the robot's location to the extracted landmarks. All the landmark vectors are averaged to form the average landmark vector, and the home vector can be obtained by subtraction between the average vector at the current location and the average vector at the home location [20–23]. HiSS adopts the SIFT features as the landmarks, and classifies all the features into contracted features and expanded features by the relation between the SIFT scale value and spatial distance. These two different types of the features can produce the landmark vectors in different directions, and the desired home vector can be generated by integrating all the landmark vectors [24, 25].

The above two landmark-based homing approaches have been widely-used in the field of visual homing. However, the precision of the extracted landmarks has a great influence on the homing performance. Since all the landmark vectors contribute equally to the home vector, the inaccurate landmarks will have a negative effect on final result [12]. Besides, both approaches need to satisfy as much as possible the equal distance assumption, which describes the optimal landmark distribution. Since visual homing only relies on the image information but no depth structure information of the environment, the distance information is difficult to obtain. To avoid the potential impact due to the lack of depth information, the equal distance assumption provides constraint that the extracted landmarks need to be distributed and located the same distance from the home

*e-mail: grady_heu@126.com

Manuscript submitted 2018-05-09, revised 2018-07-02, initially accepted for publication 2018-08-04, published in April 2019.

location [22, 23]. Although the equal distance assumption has been verified through experiments to be correct [26], obviously the optimal landmark distribution will always be violated in practical applications. Therefore, it is meaningful to ensure that the landmarks can be distributed as evenly as possible during the movement of the robot.

To cope with the above problems, this paper presents the so-called vector pre-assigned mechanism (VPM) to improve the performance for the landmark-based homing approaches. The core idea of VPM is to assign a reasonable weight to each landmark vector. If a certain landmark vector is considered unreliable or inaccurate, VPM will assign a lower weight to this vector, weakening its effect on the final home vector. Hence, the calculated home vector can be closer to the ideal vector, so that the robot can reach the destination with a better trajectory. In addition, this paper proposes a landmark distribution optimization strategy based on the imaging principle of panoramic vision. The optimization strategy evaluates the value of the landmarks based on their positions in the image, the low-value landmarks will be similarly assigned low weights so that the actual landmark distribution can exhibit nearly the same effect as the optimal distribution.

The paper is summarized as the following sections: In Section 2, we introduce the ALV and HiSS algorithms in detail. We then present VPM and landmark optimization strategy in Section 3. In Section 4, we perform a series of experiments on panoramic image database and a real mobile robot along with the analysis. Finally, we draw conclusions and point out the future work in Section 5.

2. ALV and HiSS

2.1. ALV. The ALV model is characterized in Fig. 1. C is the robot's current location. H is the reference home location. L_i is the ith landmark, where $i = 1, 2, \ldots, n$. AL is the virtual average landmark. $\boldsymbol{h}$ is the perfect home vector pointing from C to H.

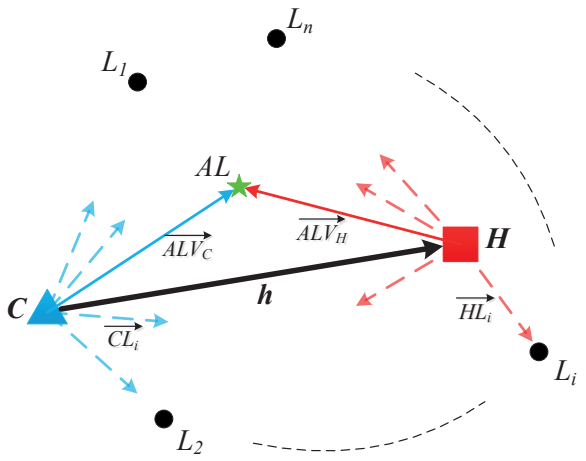


Fig. 1. The ALV model. The blue and red dotted arrows respectively denote the unit landmark vectors at C and H. The blue and red solid arrows respectively denote the average landmark vectors at C and H

We take the view of the robot at C as an example. Based on the imaging principle of panoramic vision, C is in the center of the generated panoramic image. We define C to be the origin and establish the two-dimensional Cartesian image coordinate system. The image coordinate of L_i is denoted as $T_i = (x_i, y_i)$. Thus, the unit landmark vector of L_i can be expressed by:

$$\overrightarrow{CL_i} = \frac{T_i - T_C}{\|T_i - T_C\|}. \tag{1}$$

Where $T_C = (0, 0)$ is the coordinate of C. For all the n landmarks extracted at C, the average landmark vector can be calculated by:

$$\overrightarrow{ALV_C} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{CL_i}. \tag{2}$$

By the same token, the average vector at H can be computed as follows:

$$\overrightarrow{ALV_H} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{HL_i}. \tag{3}$$

Finally, the unit home vector $\boldsymbol{h}$ can be generated by the subtraction of the ALV at H and the ALV at C:

$$\boldsymbol{h} = \frac{\overrightarrow{ALV_C} - \overrightarrow{ALV_H}}{\left\|\overrightarrow{ALV_C} - \overrightarrow{ALV_H}\right\|} = (\cos\theta, \sin\theta), \quad \theta \in [0, 360°). \tag{4}$$

Where θ is the angle between the home vector and the positive x-axis of C coordinate system.

As a conceptually simple homing strategy, ALV only relies on vector calculation to compute the desired home vector with extremely fast speed. However, ALV is highly dependent on the precision of the landmarks. In essence, the ALV algorithm is equivalent to simplifying all the landmarks into an average landmark (i.e. AL in Fig. 1), and the home vector is derived from the difference in perspective of the robot when it observes the average landmark at the current and home location. The original ALV algorithm averages all the landmarks directly without considering their precision in terms of locations. Therefore, inaccurate landmarks will greatly affect the home vector, especially when the total number of the landmarks is limited. Detecting inaccurate landmarks and reducing their bad impact is the most efficient solution to improve the performance of ALV.

2.2. HiSS. The original HiSS algorithm uses SIFT features as landmarks because SIFT is a reliable local feature detection and description algorithm. SIFT is invariant to rotation, affine and illumination, and it also has a crucial phenomenon adopted by HiSS: the scale value of the SIFT feature is negatively correlated with the spatial distance between the landmark and the robot. That is to say, the qualitative relation between the robot and the landmarks can be determined by the scale space of

SIFT. When the robot approaches a certain landmark, the corresponding SIFT scale value will become larger. Based on the different scale values of the two SIFT features in the same matching pair, HiSS classifies all the features in the current image into contracted and expanded features, which can further produce landmark vectors with different orientations. It is worth mentioning that HiSS is not limited to use SIFT features, the features with scale parameter can also be applied, such as SURF features.

Assuming that the two images are correctly matched to produce a SIFT feature matching pair $\left(f_i^H, f_j^C\right)$, where f_i^H is the ith feature in the goal image and f_j^C is the jth feature in the current image. We define $\Delta\sigma$ as the scale difference between f_i^H and f_j^C:

$$\Delta\sigma = \sigma_i^H - \sigma_j^C. \tag{5}$$

Where σ_i^H and σ_j^C are the scale values of f_i^H and f_j^C, respectively. HiSS classifies the features in the current image based on the following two principles:

1. If $\Delta\sigma > 0$, it can be indicated that the distance from the landmark to the current location is greater than the distance to the home location. Therefore, HiSS defines the corresponding SIFT feature as the contracted feature, the robot should move towards the features that have contracted.
2. If $\Delta\sigma < 0$, it can be indicated that the distance from the landmark to the current location is less than the distance to the home location. Therefore, HiSS defines the corresponding SIFT feature as the expanded feature, the robot should move away from the features that have expanded.

The HiSS model is shown in Fig. 2. $f_1, f_2, \ldots, f_8$ are the SIFT features, where $f_1 \sim f_4$ are the contracted features, $f_5 \sim f_8$ are the expanded features. $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_8$ are the unit landmark vectors corresponding to the SIFT features. μ is the perpendicular bisector of $\overline{CH}$, and μ' is a parallel line through C.

According to the principles of HiSS, the orientations of the landmark vectors are determined based on the scale values, and the home vector can be obtained by integrating all the landmark vectors:

$$\boldsymbol{h} = \frac{1}{\left\|\sum_{i=1}^{n} \boldsymbol{v}_i\right\|} \sum_{i=1}^{n} \boldsymbol{v}_i. \tag{6}$$

Where n is the total number of the landmark vectors.

HiSS is a trend-oriented visual homing approach, the calculated landmark vectors do not point directly to the destination, but rather point in the direction that will guide the robot closer to the destination. It turns out that HiSS can guide the robot to the destination gradually with a high degree of accuracy, but it still has two drawbacks. The first drawback is the same as that of ALV, HiSS also has high requirements for the accuracy and distribution of the landmarks. The second drawback has not yet been resolved, that is, the directions of all the landmark vectors in Region B are completely opposite to the desired directions. Based on the principle 2 of HiSS, the features in Region B will be defined as the expanded features. However, each calculated

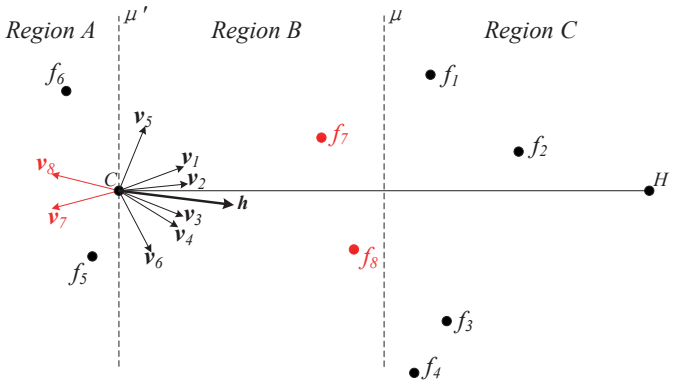


Fig. 2. The HiSS model

landmark vector in Region B is obviously considered to be a 'bad vector', which has a 180° angular error from the desired vector ($\boldsymbol{v}_7$ and $\boldsymbol{v}_8$ in Fig. 2). Although Region B is small in size relative to the entire environment, the 'bad vectors' can still have a negative effect on the result.

3. Proposed methods

The proposed methods include VPM and the landmark optimization strategy. VPM contains two solutions, called VPM-ALV and VPM-HiSS, which respectively optimize the homing performance of ALV and HiSS.

3.1. VPM-ALV. VPM-ALV (abbreviated as V-ALV) optimizes the computational process of the original ALV algorithm, it first calculates the home sub-vectors and evaluates their contribution, then assigns a specific weight to each sub-vector. For the low-quality home sub-vectors identified by VPM, the assigned weights of them will be reduced, the sub-vectors with too low contribution will even be eliminated directly. Finally, the desired home vector can be generated by integrating all the weighted home sub-vectors.

The detailed calculation process of V-ALV can be described as follows. For the ith landmark L_i in the scene, the corresponding home sub-vector $\boldsymbol{h}_i$ can be calculated by:

$$\boldsymbol{h}_i = \frac{\overrightarrow{CL_i} - \overrightarrow{HL_i}}{\left\|\overrightarrow{CL_i} - \overrightarrow{HL_i}\right\|} = (\cos\theta_i, \sin\theta_i), \quad \theta_i \in [0, 360°). \tag{7}$$

Where θ_i is the angle between $\boldsymbol{h}_i$ and the positive x-axis. To assign a specific weight to each home sub-vector, V-ALV establishes a surrounding space of C, classifying all the home sub-vectors based on their different orientations. The surrounding space of V-ALV is characterized in Fig. 3, the space is equally divided into eight sector units, denoted as $V_1, V_2, \ldots, V_8$. V-ALV assigns the weight to each home sub-vector based on the number of the vectors contained in each sector unit.

Facts have proved that if the angular error between the calculated and desired home vector is less than 90°, the robot can

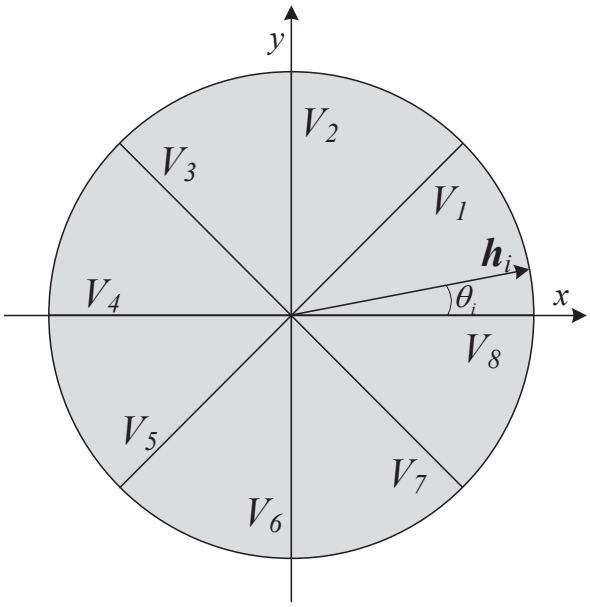


Fig. 3. Surrounding space and sector units of V-ALV

eventually reach the home location [24]. Hence, according to the distribution of the sub-vectors in the surrounding space, V-ALV classifies all the sectors as major, secondary and unrelated sectors to perform the weight assignment. Assuming that w_i is the weight of $\boldsymbol{h}_i$, the sector classification and weight assignment rules are stated as follows:

1) Major sector: V-ALV considers the sector containing the largest number of the vectors as the major sector, denoted as V_m. We define n_m as the number of the vectors included in V_m, and the weights of all these n_m sub-vectors are set to 1:

$$w_i = 1, \ \ if \ \ \boldsymbol{h}_i \in V_m. \tag{8}$$

2) Secondary sector: The secondary sector is defined as the sector less than 90° from V_m. In other words, this type of the sector should be determined from the four candidate sectors closet to V_m. V-ALV assumes that if the number of the home sub-vectors in a candidate sector is not less than 75% of n_m, this sector is determined as the secondary sector, denoted as V_s. We define n_s as the number of the vectors included in V_s, and the weights of these vectors are set to the ratio of n_s to n_m:

$$w_i = \frac{n_s}{n_m}, \ \ if \ \ \boldsymbol{h}_i \in V_s. \tag{9}$$

3) Unrelated sector: If a sector is located more than two sector units away from V_m, or if the number of the vectors it contains is less than 75% of n_m, this sector is determined as the unrelated sector, denoted as V_u. The weights of all the vectors in V_u are set to 0, which means that these vectors are no longer involved in subsequent calculations:

$$w_i = 0, \ \ if \ \ \boldsymbol{h}_i \in V_u. \tag{10}$$

After each home sub-vector has been assigned a weight, the final home vector can be computed by summing up all the weighted sub-vectors:

$$\boldsymbol{h} = \frac{1}{\left\| \sum_{i=1}^{n} w_i \boldsymbol{h}_i \right\|} \sum_{i=1}^{n} w_i \boldsymbol{h}_i = (\cos\theta, \sin\theta). \tag{11}$$

A pseudo-code description of V-ALV is given in Algorithm 1.

Algorithm 1. Pseudo-code of V-ALV.

```
1: V_i is set as the ith sector unit of the surrounding space
2: n_i is set as the number of the home sub-vectors in V_i
3: n_m ← max (n_1, n_2, …, n_8)
4: Determine the sector containing n_m vectors as V_m
5: Calculate the weights of vectors in V_m according to Equation (8)
6: for i ← 1 to 8 do
7:     Check whether V_i is V_m
8:     if FALSE then
9:         Check whether n_i is less than 0.75*n_m
10:        if TRUE then
11:            Determine V_i as V_u
12:            Calculate the weights of vectors in V_i according to
               Equation (10)
13:        else
14:            Check whether V_i is one of the four candidate sectors
               closet to V_m
15:            if TRUE then
16:                Determine V_i as V_s
17:                Calculate the weights of vectors in V_i according to
                   Equation (9)
18:            else
19:                Determine V_i as V_u
20:                Calculate the weights of vectors in V_i according to
                   Equation (10)
21:            end
22:        end
23:    end
24: end
25: Calculate the home vector h according to Equation (11)
```

V-ALV is a static optimization mechanism, all the home sub-vectors in the surrounding space are given specified weights based on their distribution in the sector units. The essence of V-ALV is to relate the precision and distribution of home sub-vectors. Since the home vector is derived from summing all the home sub-vectors, it can be inferred that most sub-vectors will have similar directions to the home vector. By analyzing the distribution of sub-vectors in the surrounding space, V-ALV can detect the sub-vectors that differ greatly from the directions of most sub-vectors, and these vectors are considered 'bad vectors', which may affect the homing accuracy.

3.2. VPM-HiSS. VPM-HiSS (abbreviated as V-HiSS) adopts the similar surrounding space to V-ALV, but utilizes a dynamic scan sector unit, denoted as E_{ds}. Figure 4 shows the surrounding space and E_{ds} of V-HiSS. E_{ds} is a semicircular sector rotating around the center of the surrounding space.

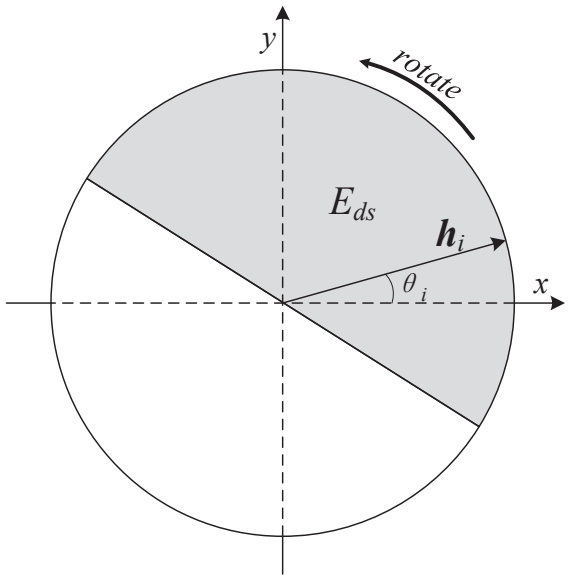


Fig. 4. Surrounding space and sector unit of V-HiSS

After all the landmark vectors are generated by the original HiSS algorithm, some vectors will be covered in E_{ds}, and the others will fall outside E_{ds}. During the rotation of E_{ds}, the total number of vectors that lie in E_{ds} can be computed continuously. When we obtain the maximum number of vectors in E_{ds} (larger than any rotating step), the optimal location of E_{ds} can be finally determined. Then, V-HiSS equally divides E_{ds} into four sub-sectors with 45° central angle, denoted as E_1, E_2, E_3 and E_4.

Assuming that the total number of the vectors in E_{ds} is n_{ds}, the number of the vectors in each sub-sector is n_1, n_2, n_3 and n_4, separately. The weight of the ith landmark vector in E_{ds} can be computed by:

$$w_i = \frac{n_j}{n_{ds}}, \quad if \quad w_i \in E_j. \tag{12}$$

Where $j \in \{1, 2, 3, 4\}$. For the remaining landmark vectors located outside E_{ds}, their vectors are set to 0, which means that these vectors are no longer involved in subsequent calculations:

$$w_i = 0, \quad if \quad w_i \notin E_{ds}. \tag{13}$$

Finally, the home vector can be calculated by:

$$\boldsymbol{h} = \frac{1}{\left\| \sum_{i=1}^{n} w_i \boldsymbol{v}_i \right\|} \sum_{i=1}^{n} w_i \boldsymbol{v}_i = (\cos\theta, \sin\theta). \tag{14}$$

A pseudo-code description of V-HiSS is given in Algorithm 2.

Algorithm 2. Pseudo-code of V-HiSS.

n_0 is set as the number of the landmark vectors in E_{ds} in the initial state
β is set as the angular at which E_{ds} rotates
n_β is set as the number of the landmark vectors in E_{ds} when E_{ds} rotates by β degrees
$n_{ds} \leftarrow n_o$
Equally divide E_{ds} in the initial state into four sub-sectors
Calculate the weights of vectors in each sub-sector according to Equation (12)
Calculate the home vector $\boldsymbol{h}$ according to Equation (14)
for $\beta \leftarrow 1$ to 360 **do**
    Check whether n_β is greater than n_{ds}
    if TRUE **then**
        $n_{ds} \leftarrow n_\beta$
        Equally divide E_{ds} in the current state into four sub-sectors
        Calculate the weights of vectors in each sub-sector according to Equation (12)
        Calculate the home vector $\boldsymbol{h}_\beta$ according to Equation (14)
        $\boldsymbol{h} \leftarrow \boldsymbol{h}_\beta$
    end
end

V-HiSS is a dynamic optimization mechanism, the dynamic scan sector can be freely adjusted based on the distribution of the landmark vectors. When E_{ds} is finally determined, its linear part is equivalent to μ'. All the landmark vectors located inside E_{ds} can provide a positive impact on the home vector, whereas the vectors located outside E_{ds} are exactly the opposite. Hence, V-HiSS not only effectively detects and eliminates the inaccurate landmark vectors, but also removes all the 'bad vectors' in Region B. Thus, the overall navigation performance of HiSS can be further improved.

3.3. Landmark optimization strategy. Based on the imaging principle of panoramic vision, the landmark optimization strategy (abbreviated as LOS) is presented to optimize the distribution of the extracted landmarks. The core idea of LOS is to divide the effective area of the panoramic image into several sub-areas, and the landmarks located in different sub-areas will be assigned corresponding weights. The contribution of the landmarks located away from most landmarks will be reduced, so that the actual distribution will satisfy the equal distance assumption without any landmark elimination.

Most of the homing approaches use panoramic images as input, because this type of image usually contains the environmental information of the entire 360 degrees in horizontal direction [27], and abundant visual information from panoramic images enables the homing approaches to have more landmarks.

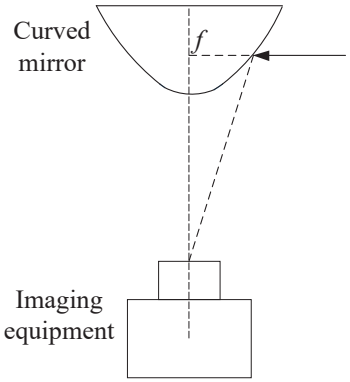


Fig. 5. Simplified model of panoramic imaging system

Fig. 5 shows the simplified model of the panoramic imaging system, which includes a curved mirror and an imaging equipment. The curved mirror reflects the real scene onto the imaging equipment, forming the panoramic image.

A crucial imaging principle for the panoramic vision image is as follows: For the scenes with the same height as the focus of the curved mirror, their projected pixels must be on a fixed circle in the image, called horizon circle. No matter where the panoramic imaging system moves, as long as its vertical height does not change, these pixels will remain on the horizon circle.

Inspired by the above principle, we can conclude that the extracted landmarks nearby the horizon circle are considered to be more stable and reliable for visual homing. On one hand, as the robot moves, the projected pixels of these landmarks will always be near the horizon circle without a large shift in position, so that the overall distribution can remain stable. On the other hand, taking into account the actual height of the panoramic imaging system, the landmarks nearby the horizon circle always contain rich visual information. However, the landmarks away from the horizon circle tend to represent higher or lower parts of the environment, such as indoor ceiling (or outdoor sky) and floor. The visual information of these landmarks is poor with relatively higher mismatching rate.

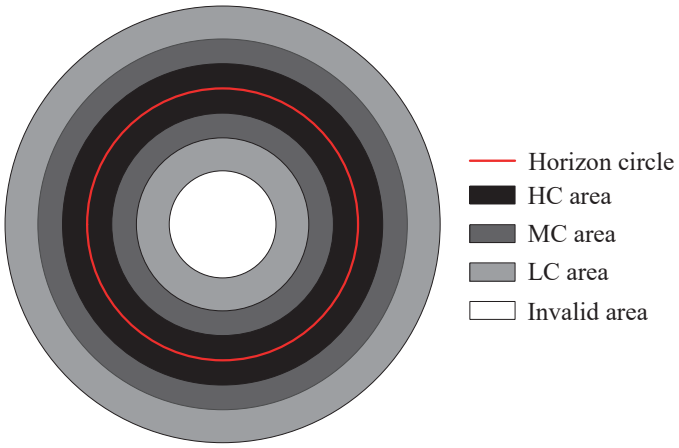


Fig. 6. The model of LOS

The model of LOS is shown in Fig. 6. LOS divides the valid area of the original panoramic image into five sub-areas, including one high-contribution (HC) area, two medium-contribution (MC) areas and two low-contribution (LC) areas. LOS assigns weights to the landmarks located in different areas so that more stable and reliable landmarks can provide higher contribution. The detailed parameter settings we select for LOS are shown in Table 1. The proportion values refer to the ratio between the width of the sub-area and the total width of the valid area, each MC and LC area accounts for 15% and 20%, respectively.

Table 1
Parameter settings of LOS

Area	Proportion	Weight
HC	30%	1
MC	30%	0.75
LC	40%	0.5

After assigning the weights to the landmarks located in different areas, the above two homing approaches no longer utilize the unit landmark vectors directly, but the vectors with different modulus values. The landmarks nearby the horizon circle have a major contribution to the generation of the home vector, and the effect of the landmarks farther from the horizon circle are weakened.

Based on the parameters in Table 1, for each landmark, we can thus determine the modulus of the corresponding landmark vector as follows:

$$\begin{cases} \|\boldsymbol{v}_i\| = 1, & if \ \ \boldsymbol{v}_i \in \text{HC area} \\ \|\boldsymbol{v}_i\| = 0.75, & if \ \ \boldsymbol{v}_i \in \text{MC area} \\ \|\boldsymbol{v}_i\| = 0.5, & if \ \ \boldsymbol{v}_i \in \text{LC area} \end{cases} . \tag{15}$$

Where $i = 1, 2, \ldots, n$. Therefore, under the premise of not eliminating the landmarks, LOS reduces the weights of the low-quality landmarks, so that the actual landmark distribution can be closer to the optimal uniform distribution, improving the accuracy of the homing approaches.

3.4. Overview of the proposed method. Figure 7 shows the flow diagram of the proposed methods, including three steps as follows:

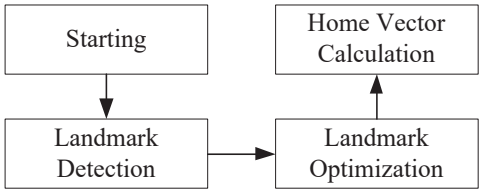


Fig. 7. Flow diagram of the proposed homing method

1) Landmark detection: We utilize SURF features as natural landmarks. SURF (speeded up robust features) is a local feature detector and descriptor, which is used to extract unique and robust image features. Compared with the classic SIFT algorithm, SURF can greatly improve the calculation speed while ensuring high feature matching accuracy [28], so SURF is more suitable for real-time robotic applications.
2) Landmark optimization: The purpose of this step is to optimize the landmark distribution to satisfy the equal distance assumption. According to Section 3.3, each extracted landmark is assigned a specific weight, generating the landmark vector with a reasonable modulus instead of a fixed unit length.
3) Home vector calculation: According to Section 3.1 and 3.2, the home vector can be obtained by the proposed V-ALV or V-HiSS.

4. Experiments

4.1. Experimental scene and robot platform. The experimental scene is shown in Fig. 8a. The experimental area for robot visual homing is a 4.5 m×3 m indoor space surrounded by different kinds of objects (such as tables, chairs, manipulator and filing cabinets). We selected 9×6 = 54 representative locations in the experimental area to test the overall performance of the homing approaches, the distance between two adjacent locations is 50 cm. The experimental area is shown in Fig. 8b.

Related experiments were carried out on an omni-directional mobile robot with panoramic imaging system. The robot is equipped with mecanum wheels, it can move in any direction without changing its orientation. The mobile robot platform is shown in Fig. 9.

A database of panoramic images was collected. We kept the current experimental environment as the default situation, the people in the environment were immobile, the curtains were open, and the overhead fluorescent lights were on. All the images were captured by positioning the robot platform at each representative location. During the collection, the condition of the environment was unmodified and the robot platform was always oriented in the same direction. The resolution of these images is 680×680, and the relevant parameters selected in Section 3 are all based on this set of images. Three panoramic image samples are shown in Fig. 10.

Fig. 9. Mobile robot platform

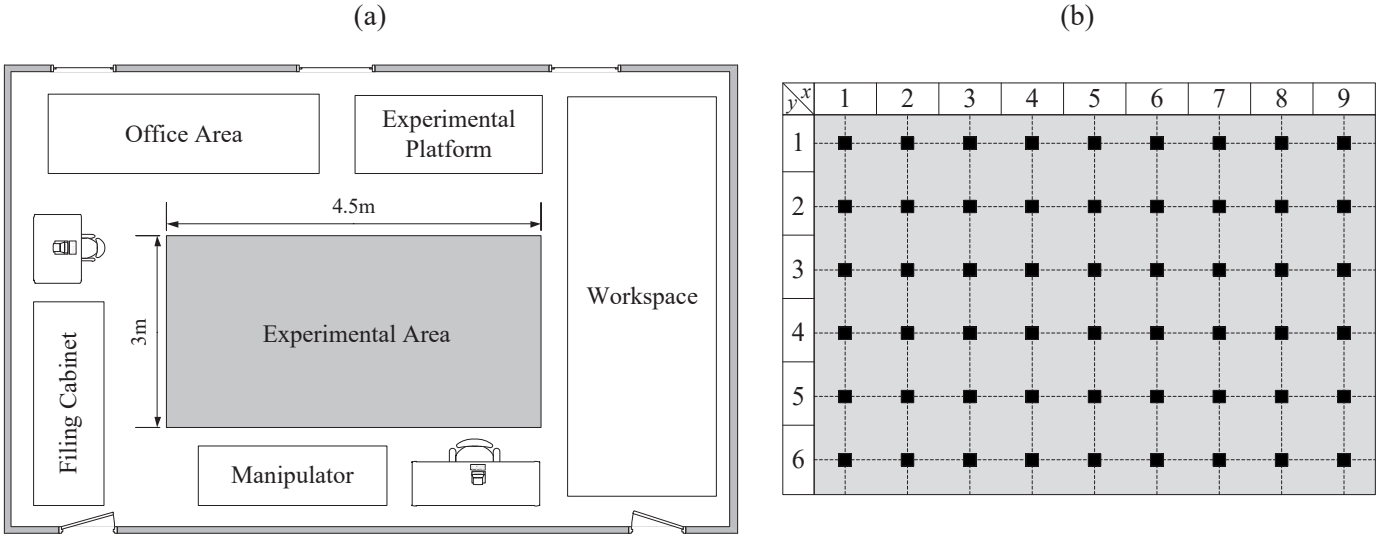


Fig. 8. Experimental scene. a) Scene layout. b) Experimental area. The black squares represent the selected representative locations

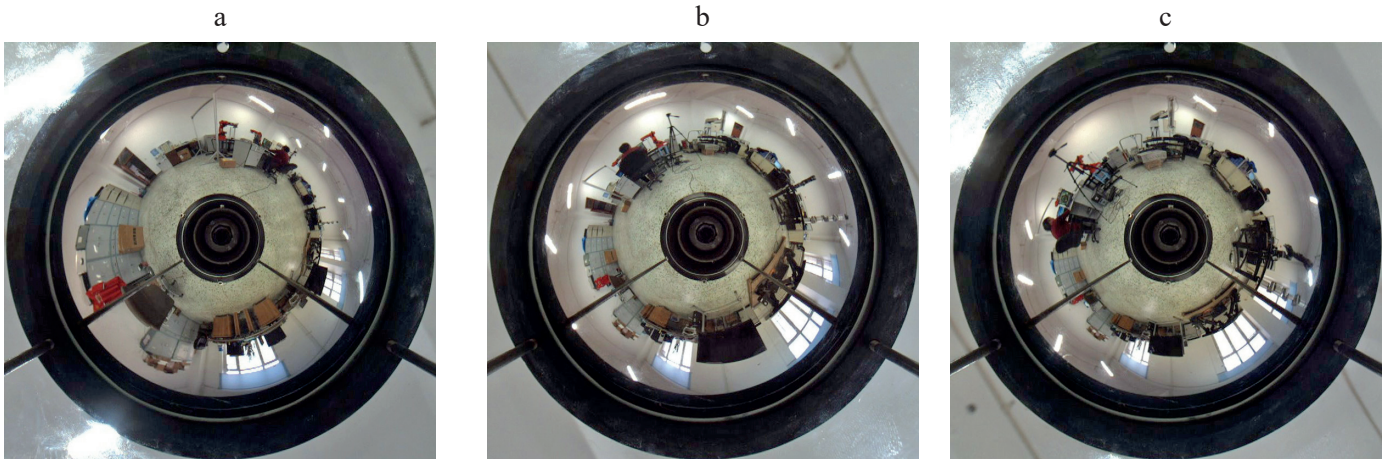


Fig. 10. Panoramic image samples collected at three representative locations. a) Location (1,3). b) Location (5,5). c) Location (8,4)

4.2. Performance metrics. Two widely used performance metrics in the field of visual homing were adopted to quantitatively evaluate the performance.

The first metric, *AAE* (average angular error), is to measure the average difference between actual homing direction α and ideal homing direction α_{ideal}. Taking C as the origin of the world coordinate system, the actual homing direction α and the ideal homing direction α_{ideal} can be defined as follows:

$$\begin{cases} \alpha = \text{atan2}\left(h_y, h_x\right) \\ \alpha_{ideal} = \text{atan2}\left(y_H - y_C, x_H - x_C\right). \end{cases} \tag{16}$$

Where $\left(h_y, h_x\right)$ denotes the calculated home vector $\boldsymbol{h}$. $\left(x_H, y_H\right)$ and $\left(x_C, y_C\right)$ respectively denote the coordinates of H and C, the angular error $AE(H, C)$ can be computed by:

$$AE(H, C) = \left|\alpha - \alpha_{ideal}\right|. \tag{17}$$

Where $AE(H, H) = 0$. For $s \times t$ representative locations, the average angular error $AAE(H, C)$ can be computed by:

$$AAE(H, C) = \frac{1}{st}\sum_{q=1}^{st} AE\left(H, C_q\right). \tag{18}$$

Where q is the index of the current locations. In our designed experiment, s is set to 9 and t is set to 6. Lower *AAE* value indicates a more precise trajectory of the robot.

The second metric, *RR* (return ratio), is to measure the number of all possible starting locations that can successfully reach the home location. At some starting location, a dummy robot is placed to carry out a simulated movement according to the calculated home vector. The robot moves at a step of r_{li}, which is the ratio between the step length of the robot and the sampling interval of the two adjacent locations. The value of r_{li} is set to 0.8 in this experiments. If the robot can continually compute the home vector and moves by r_{li} until the destination is reached, we declare the homing process successful. If the robot travels a distance longer than half the circumference of the experimental area or out of the boundary, we declare the homing process failed. For the $s \times t$ locations, we select one location as the destination and the remaining st-1 locations as the possible starting locations. By repeating the homing process from each starting location to the destination, the ratio return *RR* can be computed by:

$$RR = \frac{k}{st - 1}. \tag{19}$$

Where k refers to the total number of the starting locations that can successfully reach the destination. Higher *RR* value indicates that the robot can return to the destination from more locations.

4.3. Experiments on image database. To evaluate the actual working process of VPM, we arbitrarily selected two images from the database to perform the experiments. We set (4,4) as the current location and (6,3) as the home location, so the desired unit home vector is (0.89, –0.45). Figure 11 shows the vector distribution statistics of VPM. In total, 804 landmarks are extracted from the two images by using SURF algorithm, and VPM classifies the home sub-vectors (or landmark vectors) based on the principles in Section 3. For V-ALV, V_1 was determined as the major sector, V_2, V_3 and V_8 were determined as the secondary sectors, $V_4 \sim V_7$ were determined as the unrelated sectors. For V-HiSS, 713 landmark vectors were contained in E_{ds}.

Table 2 shows the calculated home vectors and *AE* values of the original and improved algorithms. It can be seen that the home vectors calculated by VPM are closer to the desired situation, both V-ALV and V-HiSS have smaller *AE* values than the original algorithms.

Table 2
Calculated home vectors and *AE* values.

Method	ALV	V-ALV	HiSS	V-HiSS
$\boldsymbol{h}$	(0.73, –0.68)	(0.77, –0.64)	(0.81, –0.58)	(0.86, –0.51)
***AE*(°)**	16.22	13.25	8.98	4.36

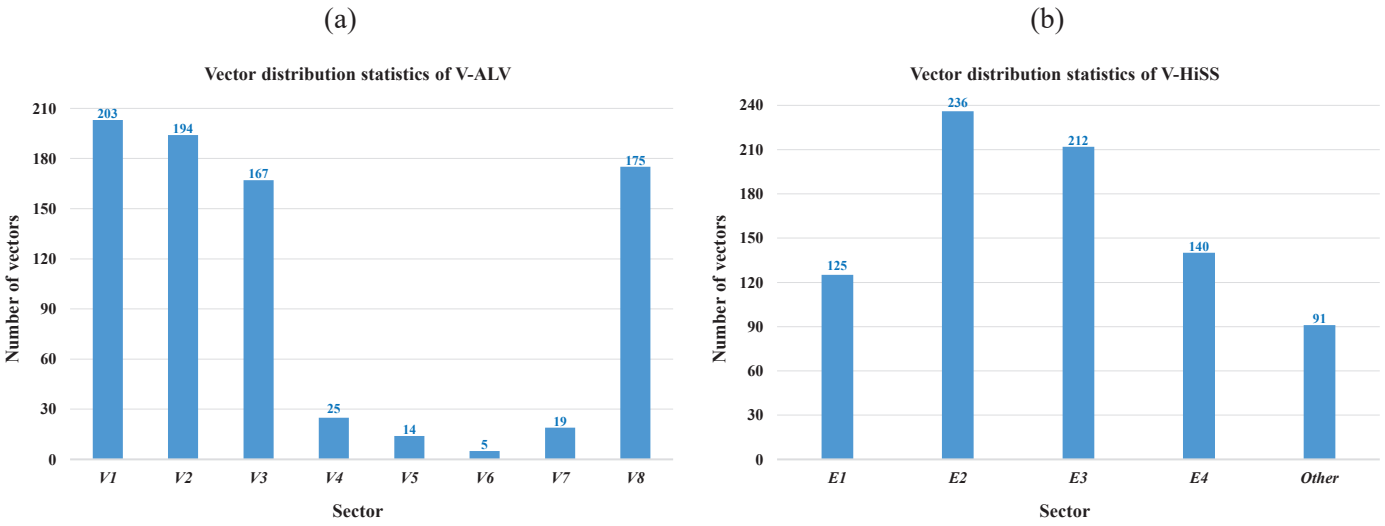


Fig. 11. Vector distribution statistics. a) V-ALV. b) V-HiSS. The '*Other*' item in the abscissa represents the area outside E_{ds}

To evaluate the overall homing performance, experiments on the image database introduced in Section 4.1 were performed. We arbitrarily selected a representative location as the home location while the remaining 53 locations as all possible current locations, and then the home vector was calculated by the homing approaches from each current location to the home location. The overall visualization results are called home vector field. Figure 12 shows the home vector fields of different homing approaches, lo-

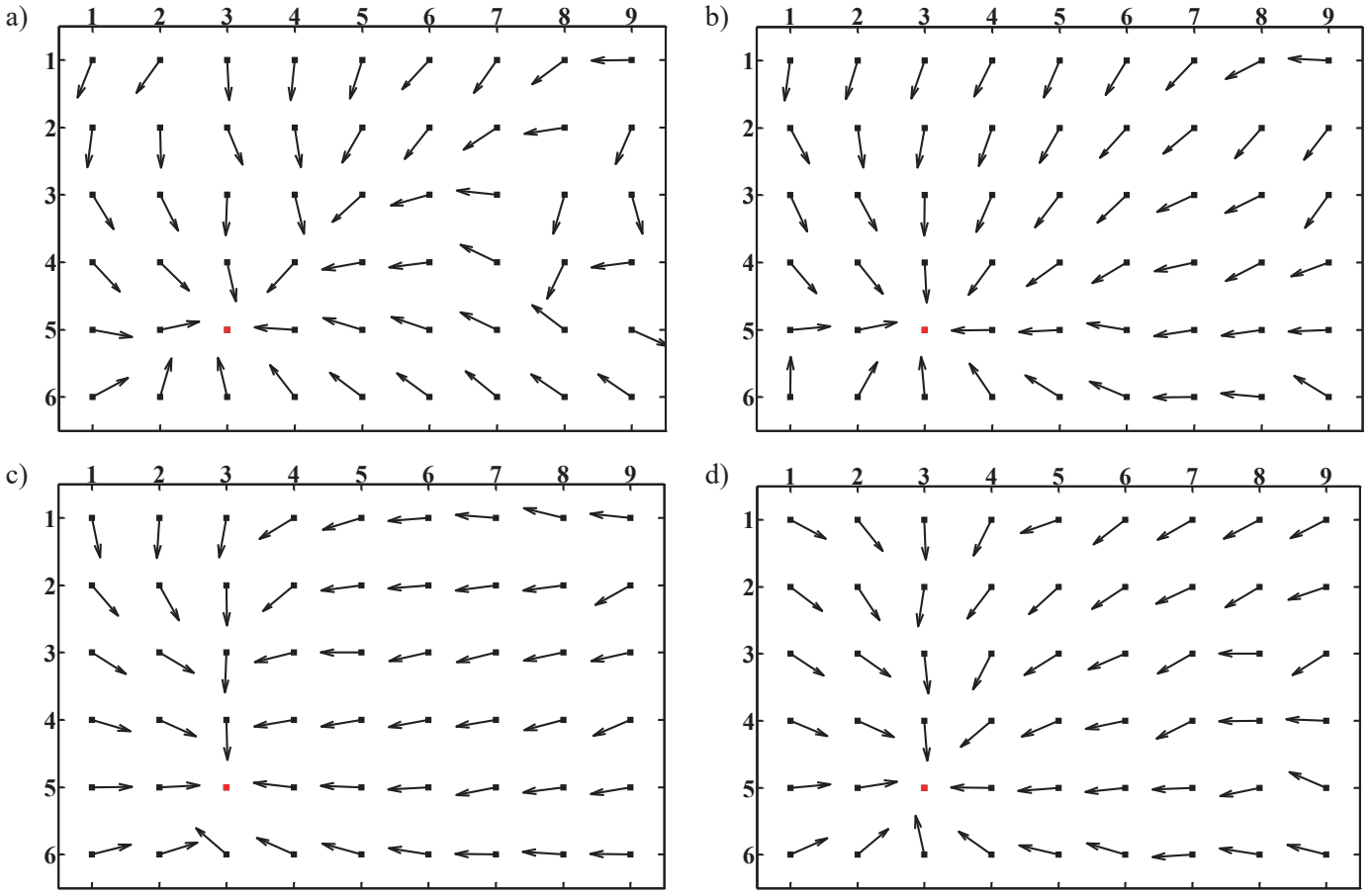


Fig. 12. Home vector fields of the homing approaches. The red squares represent the home location (3,5). The arrows represent the calculated home directions pointing from all possible current locations to the home location. a) ALV. b) V-ALV. c) HiSS. d) V-HiSS

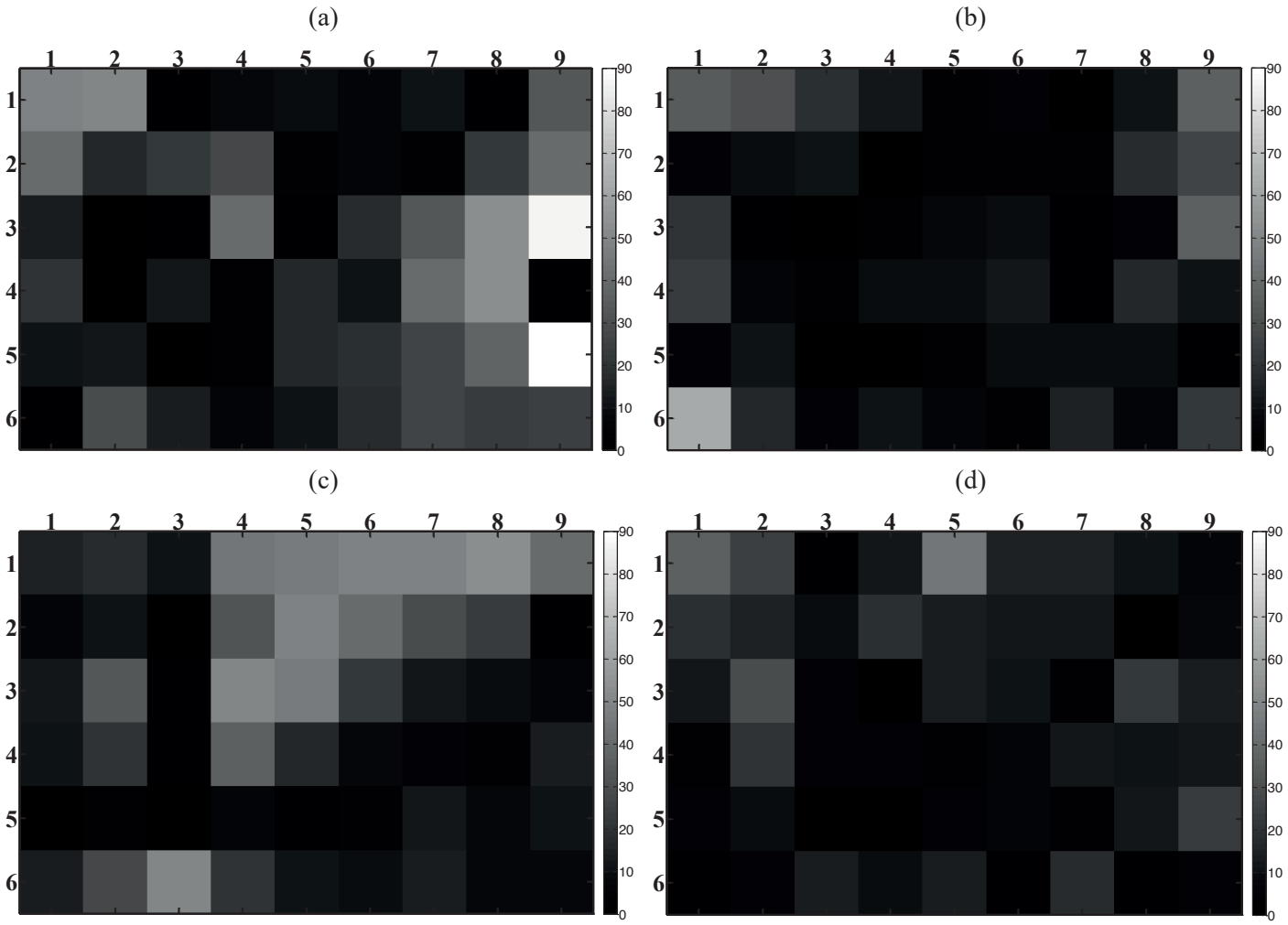


Fig. 13. *AE* grid plots with the home location (3,5). The color change of the unit block from black to white shows a gradually increasing process of *AE* from 0° to 90°. a) ALV. b) V-ALV. c) HiSS. d) V-HiSS

cation (3,5) was selected as the home location. Figure 13 shows the representation of the *AE* grids corresponding to the home vector fields, overall darker plots indicate better effects.

To achieve a more general conclusion, we performed four more database-based experiments and evaluated the related performance metrics. In addition to the location (3,5) mentioned above, we also selected (1,2), (6,4), (7,1) and (9,5) as home locations separately. These five locations are evenly and discretely distributed in the experimental area. Figure 14 and 15 shows the *AAE* and *RR* results of the homing approaches.

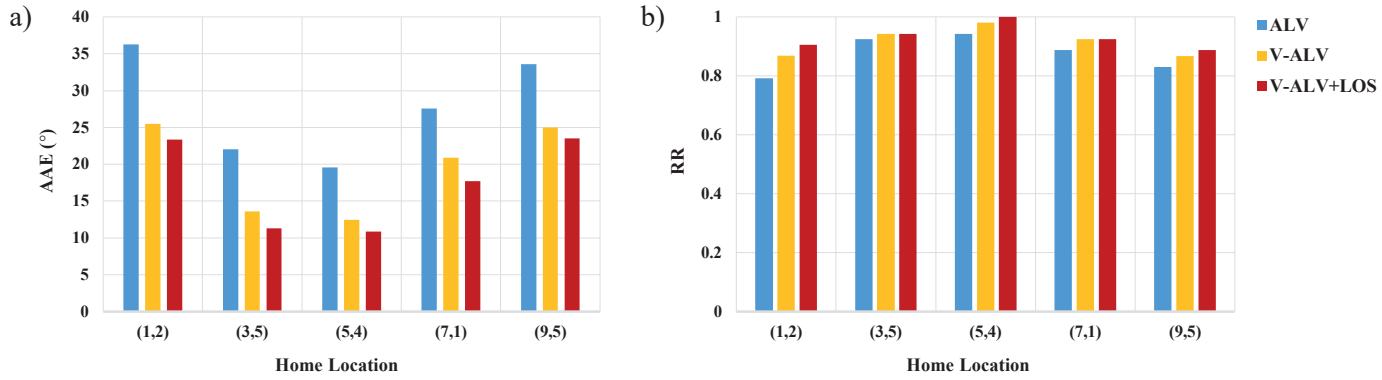


Fig. 14. *AAE* and *RR* results of ALV and V-ALV. The blue histograms represent the original ALV algorithm. The orange histograms represent the V-ALV algorithm. The red histograms represent the V-ALV algorithm with LOS. a) The *AAE* results. b) The *RR* results

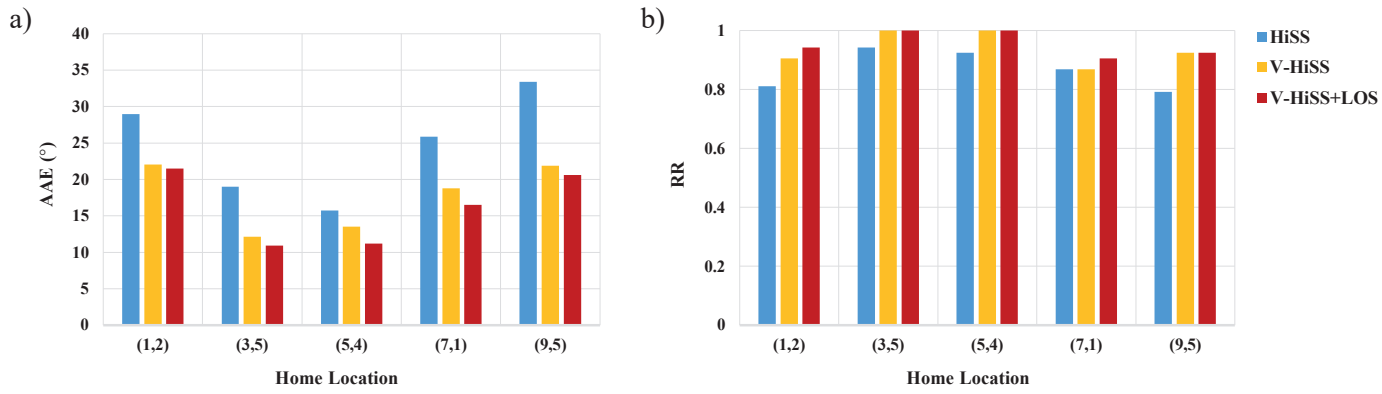


Fig. 15. *AAE* and *RR* results of HiSS and V-HiSS. The blue histograms represent the original HiSS algorithm. The orange histograms represent the V-HiSS algorithm. The red histograms represent the V-HiSS algorithm with LOS. a) *AAE* results. b) *RR* results

Table 3
Related statistics of RR values for all the 54 possible homing locations

Metrics	ALV	V-ALV	V-ALV + LOS	HiSS	V-HiSS	V-HiSS + LOS
Min	0.377	0.472	0.472	0.528	0.623	0.642
Max	1.000	1.000	1.000	1.000	1.000	1.000
Mean	0.807	0.892	0.907	0.860	0.904	0.912
Quartile (Q_1)	0.698	0.830	0.868	0.774	0.830	0.849
Quartile (Q_2)	0.830	0.962	0.981	0.868	0.962	0.962
Quartile (Q_3)	0.962	0.981	1.000	0.962	0.981	0.981

It can be seen that VPM significantly improves the homing performance. Compared with the original ALV algorithm, the overall *AAE* value of V-ALV is reduced by 29.96% and the total *RR* value is increased by 4.7%. Compared with the original HiSS algorithm, the overall *AAE* value of V-HiSS is reduced by 28.47% and the total *RR* value is increased by 8.3%. LOS further improves the homing performance according to the results. VPM with LOS is closer to the perfect home vector, the robot can reach the home location from more starting locations with better trajectories.

In addition to the above five representative home locations, we also calculated the *RR* values for all the 54 possible home locations to evaluate the overall homing success rate. For the above homing approaches, we adopted the minimum, maximum mean and quartile values of all *RR* for comparison. The related statistics are shown in Table 3, and the conclusions that can be drawn in Table 3 are the same as those in Fig. 14 and 15, robots can reach the possible home locations with higher success rates by using VPM and LOS.

Table 4 shows the average time required to calculate a home vector for different homing approaches (Inter core i7–6800 K 3.4 GHz, MATLAB R2012a). It can be seen that the time spent by the proposed approaches for computing a home vector is nearly the same as the original approaches, the performance improvement has almost no influence on the calculation speed.

Table 4
Average time required to calculate a home vector

Method	Times (s)	Method	Times (s)
ALV	0.525	HiSS	0.529
V-ALV	0.532	V-HiSS	0.551
V-ALV + LOS	0.547	V-HiSS + LOS	0.558

4.4. Experiments on a real mobile robot. To evaluate the practical homing performance, experiments on a real mobile robot were performed. We arbitrarily selected three locations as the home locations with three different starting locations for each home location. The goal image at these home location were captured and pre-stored, and the detailed experiment step is described as follows:

Step 1: The robot captures the panoramic image at the starting location.

Step 2: The robot travels 30 cm in a straight line based on the homing direction computed by different homing approaches, and then pauses.

Step 3: If one of the following three cases happens, jump to Step 5.

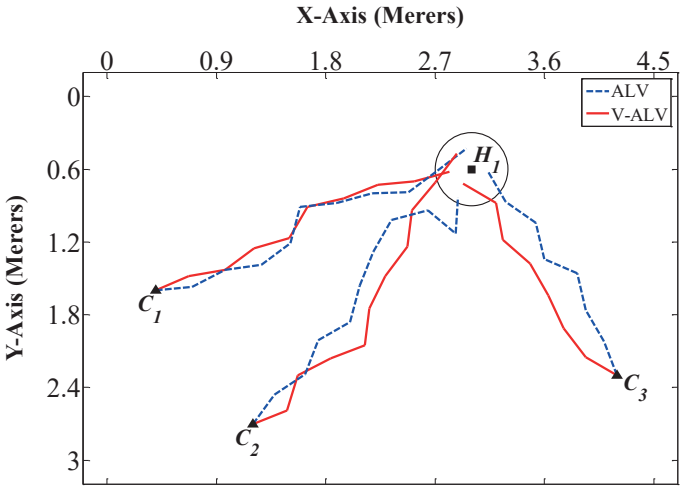
Case 1: The robot reaches a range of 30 cm centered on the home location.

Case 2: The robot moves more than 750 cm.
Case 3: The robot moves out of the experimental area.
Step 4: Continue to perform Step 1.
Step 5: If Case 1 happens, the current homing process is declared successful. If Case 2 or 3 happens, the current homing process is declared failed.
Step 6: The robot is stopped manually.

Figures 16–18 show the robot's actual trajectories guided by different homing approaches, with tables recording the *AAE* values and the total number of the robot's movement steps *N*. It can be seen intuitively that most of the trajectories obtained by the proposed approaches are smoother than those of the original approaches, the robot can reach the destination with a trajectory that is closer to a straight line. In total, 18 sets of homing tests were performed, the original ALV and HiSS algorithms have a total of two homing task failures, while the success rate of V-ALV and V-HiSS reached 100%. For the 16 sets of homing tests where both the original and improved algorithms have successfully guided the robot to the home area, the average *AAE* value for the original algorithms is 18.65°, while for the improved algorithms, the value is only 13.09°, the direction error of the robot homing is reduced by 29.81%. Both the trajectories and the statistics have proved the effectiveness of the proposed algorithms, the conclusions that can be drawn are consistent with those in Section 4.3.

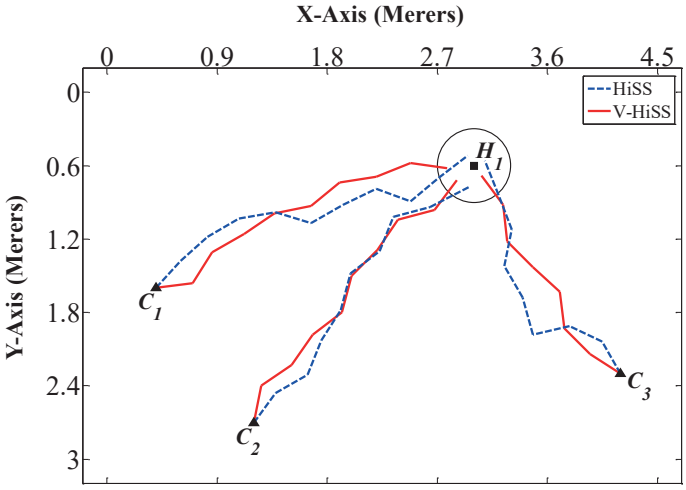
To compare the results of the simulation and actual experiments, we provided the comparison between homing trajecto-

(a)



Metrics	C_1		C_2		C_3	
	ALV	V-ALV	ALV	V-ALV	ALV	V-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes
***AAE* (°)**	17.10	9.84	23.06	21.94	22.58	19.93
N	10	9	10	10	7	7

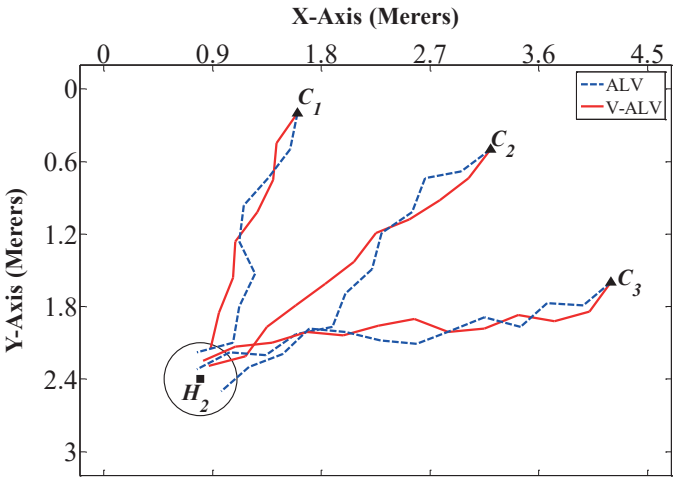
(b)



Metrics	C_1		C_2		C_3	
	HiSS	V-HiSS	HiSS	V-HiSS	HiSS	V-HiSS
Success	Yes	Yes	Yes	Yes	Yes	Yes
***AAE* (°)**	18.05	14.20	15.80	14.64	20.96	15.48
N	10	9	9	9	8	7

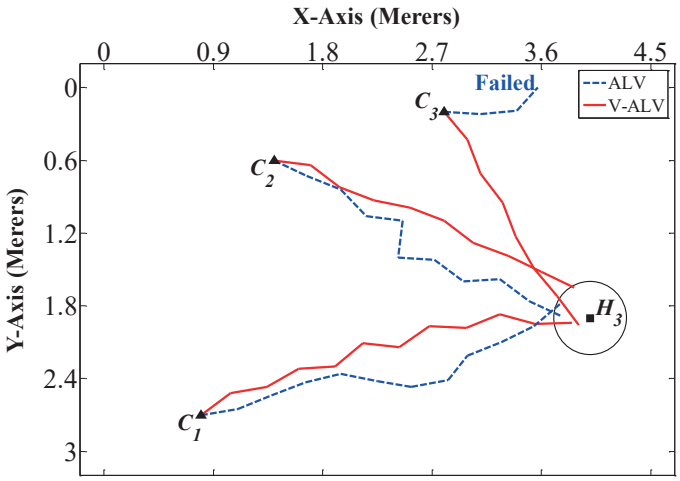
Fig. 16. Actual homing experiment 1. The home area is marked as a circle centered on a square block. The three starting locations C_1, C_2 and C_3 are marked as triangular blocks. Blue dashed lines denote the homing trajectories calculated by ALV or HiSS. Red solid lines denote the homing trajectories calculated by V-ALV or V-HiSS. Left: The robot's homing trajectories from three different starting locations; Right: Associated performance metrics. a) Results of ALV and V-ALV. b) Results of HiSS and V-HiSS

(a)



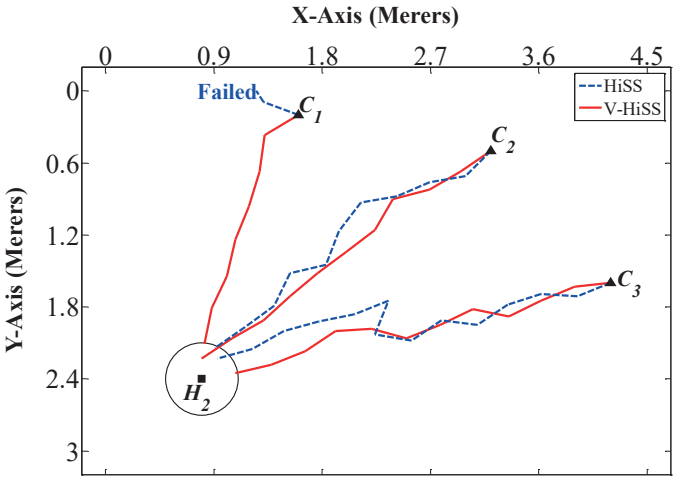
Metrics	C_1		C_2		C_3	
	ALV	V-ALV	ALV	V-ALV	ALV	V-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes
***AAE* (°)**	18.63	9.63	18.35	7.74	19.08	15.52
N	8	7	11	10	12	12

(a)



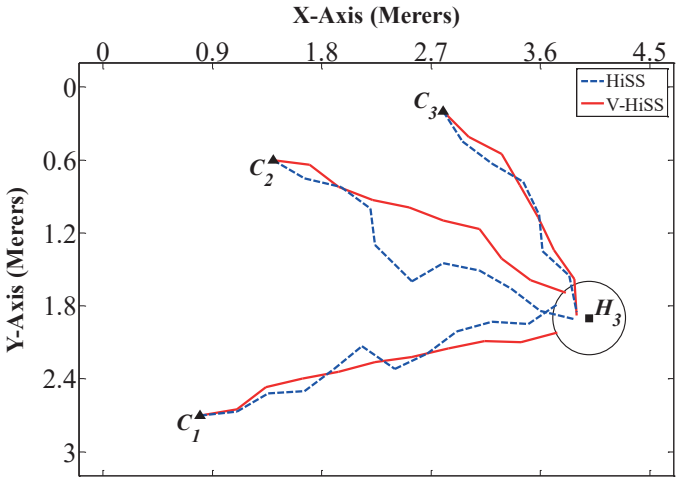
Metrics	C_1		C_2		C_3	
	ALV	V-ALV	ALV	V-ALV	ALV	V-ALV
Success	Yes	Yes	Yes	Yes	No	Yes
***AAE* (°)**	14.11	14.09	19.80	13.45	–	13.59
N	11	11	10	9	–	7

(b)



Metrics	C_1		C_2		C_3	
	HiSS	V-HiSS	HiSS	V-HiSS	HiSS	V-HiSS
Success	No	Yes	Yes	Yes	Yes	Yes
***AAE* (°)**	–	9.94	17.99	10.05	21.02	12.19
N	–	7	10	10	13	11

(b)



Metrics	C_1		C_2		C_3	
	HiSS	V-HiSS	HiSS	V-HiSS	HiSS	V-HiSS
Success	Yes	Yes	Yes	Yes	Yes	Yes
***AAE* (°)**	18.67	5.62	19.51	13.30	13.70	11.39
N	11	10	10	9	7	7

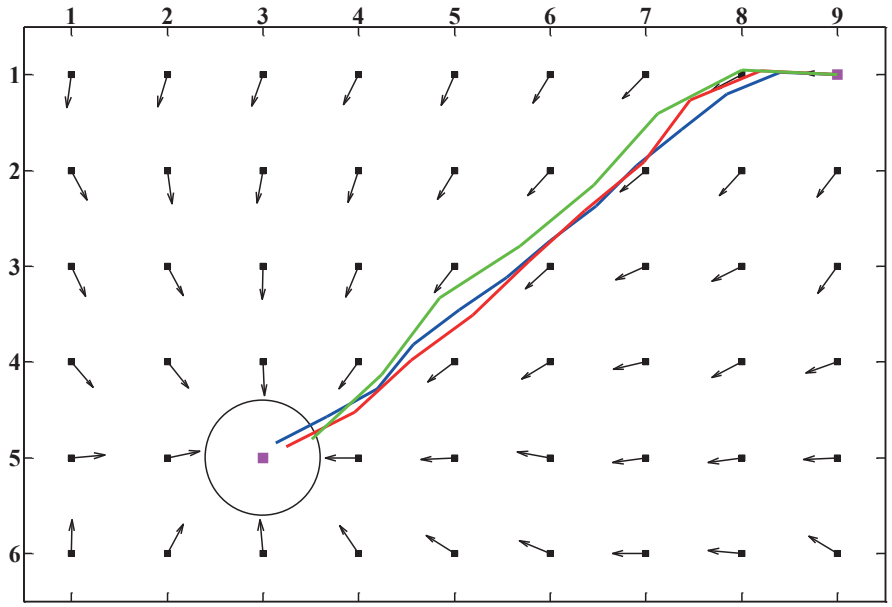
Fig. 17. Actual homing experiment 2. a) Results of ALV and V-ALV. b) Results of HiSS and V-HiSS

Fig. 18. Actual homing experiment 3. a) Results of ALV and V-ALV. b) Results of HiSS and V-HiSS

ries and home vector fields for VPM with the tables recording *AAE* and *N* values, as shown in Fig. 19 and 20. Same as the settings in Section 4.3, we still selected (3,5) as the home location, and tested the actual trajectories from the starting location (9,1) using different r_{li} values. It can be seen from Fig. 19 and 20 that the robot can successfully reach the home area based on different r_{li} values, and the actual homing trajectories of the robot basically conforms to the trend of home vector fields.

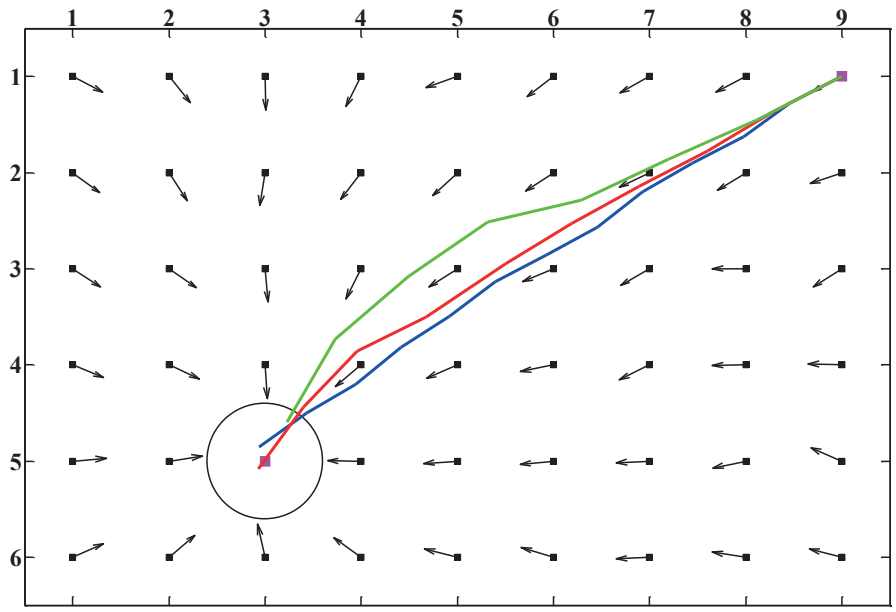
5. Conclusions

In this paper, we present an optimization mechanism for landmark-based homing approaches, called VPM. Two outstanding advantages of VPM can be summarized. On one hand, while effectively improving the homing performance, VPM has almost no influence on the calculation speed, the robot can still achieve autonomous navigation with high efficiency. On the other hand,



Metrics	$r_{li} = 0.6$	$r_{li} = 0.8$	$r_{li} = 1.0$
Success	Yes	Yes	Yes
***AAE* (°)**	7.22	10.03	11.48
N	12	9	7

Fig. 19. Comparison between homing trajectories and home vector field for V-ALV. The two magenta blocks represent the starting location (9,1) and the home location (3,5). The blue, red and green lines respectively denote the homing trajectories when r_{li} is equal to 0.6, 0.8 and 1.0. Left: The robot's homing trajectories of V-ALV based on home vector field; Right: Associated performance metrics



Metrics	$r_{li} = 0.6$	$r_{li} = 0.8$	$r_{li} = 1.0$
Success	Yes	Yes	Yes
***AAE* (°)**	7.22	10.03	11.48
N	12	9	7

Fig. 20. Comparison between homing trajectories and home vector field for V-HiSS

VPM can optimize the original algorithms by only modifying the calculation process without using other auxiliary sensors, thereby saving hardware resources. In addition to VPM, a landmark optimization strategy is proposed. The strategy optimizes the distribution of the landmarks to some extent so that the equal distance assumption is satisfied.

Related experiments were performed based on image database and a real mobile robot. The results revealed that the original ALV and HiSS algorithms can be effectively optimized by the proposed methods, the *AAE* values were reduced and *RR* values were increased, so that the robot could move from more locations with better trajectories to the specified destination. In the future, we will be more concerned with improving the performance and robustness of the visual homing methods in complex or large-scale environments, and we will also try to explore other implementations for VPM and LOS (such as adopting continuous functions or adaptive algorithms) to further improve the performance of the landmark-based homing approaches.

Acknowledgments: This work is partially supported by the National Natural Science Foundation of China (61673129, 51674109)

References

[1] J.L. Crassidis and F.L. Markley, "Three-axis attitude estimation using rate-integrating gyroscopes", *J. Guid. Control Dyn.* 39(7), 1513–1526 (2016).

[2] F. Penizzotto, E. Slawinski, and V. Mut, "Laser radar based autonomous mobile robot guidance system for groves navigation", *IEEE Latin Am. Trans.* 13(5), 1303–1312 (2015).

[3] W. Kowalczyk, M. Michałek, and K. Kozłowski, "Trajectory tracking control with obstacle avoidance capability for unicycle-like mobile robot", *Bull. Pol. Ac.: Tech.* 60(3), 537–546 (2012).

[4] M. Gupta, G.K. Arunkumar, and L. Vachhani, "Bearing only visual homing: Observer based approach", in *25th Mediterranean Conf. Control Autom. (MED)*, pp. 358–363 (2017).

[5] M. Liu, C. Pradalier, and R. Siegwart, "Visual homing from scale with an uncalibrated omnidirectional carema", *IEEE Trans. Robot.* 29(6), 1353–1365 (2013).

[6] J.O. Esparzajiménez, M. Devy, and J. L. Gordillo, "Visual EKF-SLAM from Heterogeneous Landmarks", *Sensors*, 16(4), 489 (2016).

[7] C. Gamallo, M. Mucientes, and C. V. Regueiro, "Omnidirectional visual SLAM under severe occlusions", *Robot. Auton. Syst.* 65(C), 76–87 (2015).

[8] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: A survey", *Robot. Auton. Syst.* 64(C), 1–20 (2015).

[9] N. Paramesh and D.M. Lyons, "Homing with stereovision", *Robotica*, 34(12), 2741–2758 (2016).

[10] A. Sabnis, G.K. Arunkumar, V. Dwaracherla, and L. Vachhani, "Probabilistic approach for visual homing of a mobile robot in the presence of dynamic obstacles", *IEEE Trans. Ind. Electron*. 63(9), 5523–5533 (2016).

[11] G.K. Arunkumar, A. Sabnis, and L. Vachhani, "Robust steering control for autonomous homing and its application in visual homing under practical conditions", *J. Intell. Robot. Syst*. 89(3–4), 403–419 (2018).

[12] C. Lee, S.E. Yu, and D.E. Kim, "Landmark-based homing navigation using omnidirectional depth information", *Sensors*, 17(8), 1928 (2017).

[13] Q. Zhu, X. Ji, J. Wang, and C. Cai, "A machine learning-based mobile robot visual homing approach", *Bull. Pol. Ac.: Tech.* (to be published).

[14] R. Möller, M. Krzykawski, and L. Gerstmayr, "Three 2D-warping schemes for visual robot navigation", *Auton Robot.* 29(3–4), 253–291(2010).

[15] R. Möller, M. Horst, and D. Fleer, "Illumination tolerance for visual navigation with the holistic min-warping method", *Robotics*, 3(1), 22–67 (2014).

[16] D. Fleer and R. Möller, "Comparing holistic and feature-based visual methods for estimating the relative pose of mobile robots", *Robot. Auton. Syst.* 89, 51–74 (2017).

[17] Q. Zhu, C. Liu, and C. Cai, "A novel robot visual homing method based on SIFT features", *Sensors*, 15(10), 26063–26084 (2015).

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vis*. 60(2), 91–110 (2004).

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)", *Comput. Vis. Image Underst.* 110(3), 346–359 (2006).

[20] A. Ramisa, A. Goldhoorn, D. Aldavert, R. Toledo, and R. Mantaras, "Combining invariant features and the ALV homing method for autonomous robot navigation based on Panoramas", *J. Intell. Robot. Syst*. 64(3–4), 625–649 (2011).

[21] Q. Zhu, C. Liu, and C. Cai, "A robot navigation algorithm based on sparse landmarks", *6th IEEE. Conf. Intell. Human-Machine Syst. Cybern. (IHMSC)*, pp. 188–193 (2014).

[22] Q. Zhu, X. Liu, and C. Cai, "Improved feature distribution for robot homing", *IFAC Proceedings Volumes*, 47(3), 5721–5725(2014).

[23] Q. Zhu, X. Liu, and C. Cai, "Feature optimization for long-range visual homing in changing environments", *Sensors*, 14(2), 3342–3361 (2014).

[24] D. Churchill and A. Vardy, "An orientation invariant visual homing algorithm", *J. Intell. Robot. Syst*. 71(1), 3–29 (2013).

[25] C. Lee and D.E. Kim, "Local homing navigation based on the moment model for landmark distribution and features", *Sensors*, 17(11), 2658 (2017).

[26] S.E. Yu, C. Lee, and D.E. Kim, "Analyzing the effect of landmark vectors in homing navigation", *Adapt. Behav.* 20(5), 337–359 (2012).

[27] J. Yan, L. Kong, Z. Diao et al., "Panoramic stereo imaging system for efficient mosaicking: parallax analyses and system design", *Appl. Optics.* 57(3), 396–403 (2018).

[28] J. Luo and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF", *Int. J. Image Proc.* 3(4), 143–152 (2013).