BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 66, No. 5, 2018
DOI: 10.24425/124278

A machine learning-based mobile robot visual homing approach

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

College of Automation, Harbin Engineering University, 150001, China

Abstract. Visual homing enables mobile robots to move towards a previously visited location solely based on panoramic vision sensors. In this paper, a SIFT-based visual homing approach incorporating machine learning is presented. The proposed approach can reduce the impact of inaccurate landmarks on the performance, and generate more precise home direction with simple model. The effectiveness of the proposed approach is verified on both panoramic image databases and actual mobile robot, experimental results reveal that compared to some traditional visual homing methods, the proposed approach exhibits better homing performance and adaptability in both static and dynamic environments.

Key words: robot navigation, visual homing, panoramic vision sensors, machine learning, homing performance.

1. Introduction

Autonomous mobile robot navigation has been extensively studied during the past few decades, including path planning, self localization, trajectory tracking, etc [1–3]. Visual homing is an attractive technology in this field, which provides node-to-node navigation solutions for a mobile robot to return from the current location to the reference home location. The robot compares the currently viewed image with the pre-stored goal image to generate the home vector, which describes the moving course of the robot and guides the robot to move towards the home location [4–6].

The robot visual navigation methods can be broadly partitioned into quantitative and qualitative methods, represented by on-line visual SLAM [7–9] and visual homing, respectively. The inspiration of visual homing comes from biological navigation, known as the snapshot model [10]. Compared with on-line visual SLAM, an outstanding advantage of visual homing is that it does not need any localization and mapping, but only the direction from the current location towards the destination [11, 12]. The sources of the hardware and software required for visual homing can be significantly simplified, benefitting to the effectiveness in the practical application.

In this paper, we present a novel SIFT-based visual homing approach incorporating machine learning, called machine learning-based homing (MLBH). MLBH uses linear SVM (support vector machine) classifier to help generate accurate and robust home vector. Compared with other visual homing methods, MLBH not only overcomes the constraint on the spatial distribution assumption of landmarks, but also has a high tolerance for inaccurate landmarks. Experimental results have proved that MLBH can exhibit good homing performance and adaptability in both static and dynamic environments.

The remainder of the paper is structured as follows: in Section 2, we consider the related work on visual homing. We introduce in detail the proposed MLBH method in Section 3. Then in Section 4, we carry out a series of experiments on both panoramic image databases and actual mobile robot along with an exhaustive analysis. Finally, we draw conclusions and put forward the future work in Section 5.

2. Related work

The fundamental visual homing model is illustrated in Fig. 1. All the landmarks $L_1, L_2, \ldots, L_n$ can be visible from both the robot's current location C and the desired home location H. The objective of the visual homing task is to guide the robot to move from C to H only using the information provided by these landmarks.

During the past decades, many categories of visual homing algorithms were proposed, such as warping [13], ALV [14], DID [15], HiSS [16], etc. Among them, the most representative

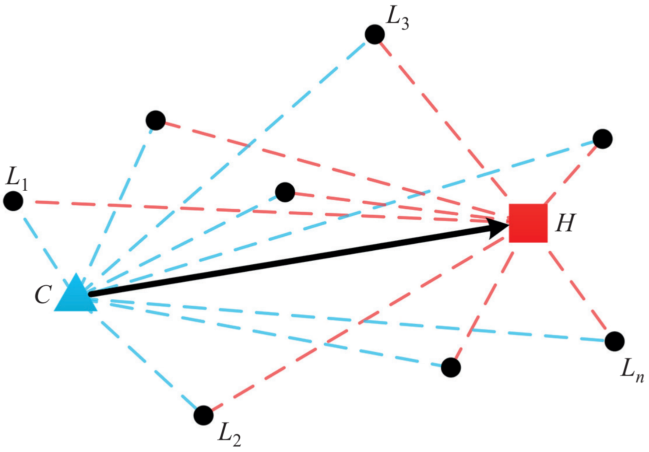


Fig. 1. Fundamental visual homing model

*e-mail: grady_heu@126.com

Manuscript submitted 2017-10-18, revised 2018-01-18, initially accepted for publication 2018-01-30, published in October 2018.

ones are warping and ALV. These two categories are widely applied in the field of visual homing, and various advanced methods involved with them have been proposed.

The warping methods distort either of the images (usually the goal image) according to the parameters describing the direction, rotation, distance of the robot's movement. By exhaustively searching for the parameters, the optimal parameter set is determined when the warped image best fits the current image, and in what follows the home vector is obtained. In addition to the original 1D-warping model, many advanced methods were presented to further optimize the performance. For example, Möller et al. proposed the 2D-warping and min-warping model by a variation of the alignment angle estimation in the environment to replace the external compass in 1D-warping [17]. Zhu et al. adopted the SIFT features as the landmarks instead of the image pixels to improve the illumination robustness [18]. A SIMD implementation and image distance measures were respectively utilized to improve the efficiency and tolerance of min-warping [19, 20]. Recently, several robotic experiments were implemented to compare the warping methods with other methods in many aspects (such as calculation speed, illumination tolerance, etc.) [21, 22].

The ALV (average landmark vector) model has been extensively used as a benchmark in visual homing. ALV is represented as an average bearing of all the possible landmarks with respect to a certain location. The robot's reliable homing behavior is derived from the difference between the ALV at the current location and the ALV at the home location. Based on the benchmark model, researchers have enrich ALV in many aspects, such as optimizing landmark distribution [23–25], quantifying keypoint distance [26] and exploiting auxiliary sensors [27]. These advanced methods lead to higher-quality landmarks or richer visual information, improving the homing accuracy and robustness greatly. Here we introduce in detail an optimized ALV method based on sparse landmarks [25] (we abbreviate the method as SL-ALV), because we use it as a comparative method with our approach. SL-ALV adopts more reasonable landmarks based on the imaging principle of panoramic vision, which is shown in Fig. 2. For landmarks that are the same height as the focus of the curved mirror, their projection pixels must be on a circle of the image, called horizon circle. A crucial property of the pixels on the horizon circle is that wherever the robot moves, these pixels will always stay on the circle. Therefore, the features nearby the horizon circle can be considered more stable and reliable for visual homing. In SL-ALV, the horizon circle is properly expanded to form a ring area, and the features are only extracted in this area instead of the whole image. The novel feature input form not only enables the low-quality features to be eliminated, but also better satisfies the equal distance assumption [13], which is the optimal distribution of the landmarks for visual homing. In summary, SL-ALV can effectively improve the homing precision and reduce the calculation amount.

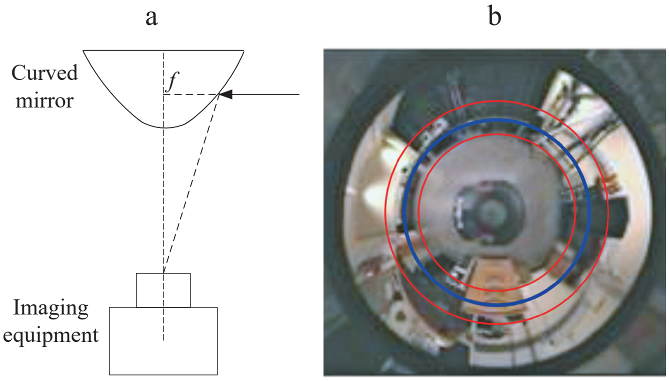


Fig. 2. Simplified model of panoramic vision system and imaging plane: a) The model of panoramic vision system containing curved mirror and imaging equipment. b) Imaging plane. The blue circle represents the horizon circle, the area between two red circles is the selected ring area

The homing approaches introduced above have been verified with great effectiveness, but most of them should obey a strict assumption on the spatial distribution of the landmarks, called equal distance assumption: The optimal homing performance can be exhibited when all the landmarks extracted from the image are uniformly distributed and located approximately the same distance from the home location [17]. However, this assumption is always unrealistic and violated, as the non-uniform distribution of landmarks will affect the performance greatly. Besides, since the contribution of each landmark to the home vector is equivalent in most homing methods, inaccurate landmarks will also impact on the precision seriously. The above problems are especially evident when dynamic variations exist, such as moving objects, illumination variations, etc. Several strategies have been presented to deal with the dynamic problems. Liu et al. used the robot's odometer information coupled with the initial visual information to develop an indoor topological navigation framework [28]. Sabnis et al. presented the bit encoding approach to extract coarse location information in probabilistic framework [29]. These strategies showed good adaptability, but wasted more sources and increased the computational complexity.

The proposed MLBH approach in this paper shows good robustness in dynamic environments. The model of MLBH is simple and easy to implement. MLBH uses the SIFT features as natural landmarks, which are divided into two categories based on their scale information. Through linear SVM Classifier, the decision boundary of the two categories are generated to help calculate the home vector. The solutions to deal with the dynamic circumstances in MLBH are mainly motivated by the merits of SVM, it can effectively reduce the impact of the inaccurate landmarks in dynamic environments, so that the precision and robustness can be improved.

3. Machine learning-based homing

3.1. Landmarks and SIFT features. We use the SIFT features as the landmarks. Due to the ground-breaking work of SIFT, the extracted features are invariant to affine distortion, addition of noise, and change in illumination [30]. By using scale space, orientation histograms and 128-dimensional keypoint descriptors, the generated SIFT features are considered to be highly distinctive so that the same features in different images can be correctly matched with good probability.

Using David Lowe's SIFT implementation available from http://www.cs.ubc.ca/~lowe/keypoints/, the ith SIFT feature f_i can be extracted and characterized as follows:

$$f_i = \left(f_{i,x}, f_{i,y}, f_{i,o}, f_{i,}\sigma, f_{i,d}\right) \tag{1}$$

where $f_{i,x}$ and $f_{i,y}$ are the coordinates of f_i in the image. $f_{i,o}$ is the orientation. $f_{i,}\sigma$ is the scale value. $f_{i,d}$ is the descriptor. For different SIFT features, these parameters cannot be exactly the same.

3.2. Qualitative relation between scale and distance. A noteworthy phenomenon are stated in [16] that the SIFT scale value is negatively correlated with the distance between the robot viewer and the natural landmark. In other words, when the robot moves away from a certain natural landmark, the corresponding SIFT scale value will become smaller. In this section, we will theoretically analyze this phenomenon.

Scale-space theory is a framework for handling image structures at different scales, it is also the prerequisite of SIFT algorithm. To build the scale-space pyramid, the initial image is repeatedly convolved with the Gaussian function and down-sampled, forming a series of smoothed images. Inspired by biological vision theory, scale space simulates the imaging process of the scene in the retina of the mammal from near to far. As the standard deviation σ (i.e. $f_{i,\sigma}$) in the Gaussian function increases, the image becomes more and more blurry. In a sense, it is equivalent to the imaging process of the scene in an actual viewer's retina when the viewer moves farther and farther away from the scene.

SIFT aims to detect and describe local features in the scale space. Due to its property of scale invariance, a certain SIFT feature can be searched and selected by detecting the local extrema at its optimal scale layer, and σ stands for the degree of blur. This process is equivalent that the viewer moves away from the corresponding landmark until an optimal position is determined, where the most reliable visual effect of the landmark can be obtained. Thus, it can be indicated that σ characterizes the distance of the viewer's movement as well. We define the distance between the certain natural landmark and the optimal position as the optimal virtual distance, for the viewer does not actually move to capture the scene images. Since each SIFT feature has only one local extrema response, the optimal virtual distance of each natural landmark must be unique.

When two images are matched, the two SIFT features in the same matching pair represent the descriptions of the same landmark from two different perspectives. Based on the above analysis, we can thus obtain the relative distance relation between the landmark and the two viewing positions by comparing the σ values of the two SIFT features: If one of the SIFT features has a larger σ value, it means that SIFT has done more Gaussian blur to detect this feature. That is to say, the viewer has moved a longer virtual distance. Due to the uniqueness of the optimal virtual distance for each landmark, it can be inferred that this viewing position is closer to the landmark.

To facilitate the interpretation, a simulated scene with simplified locations is shown in Fig. 3. Let L_k be the kth natural landmark. d_i^C and d_j^H separately stand for the distance from L_k to the current location C and the home location H, where i and j are the indexes of the features in the two images. d_v is the optimal virtual distance of L_k. We assume that L_k can be visible from both C and H, the mapped SIFT features of L_k are denoted as f_i^C and f_j^H.

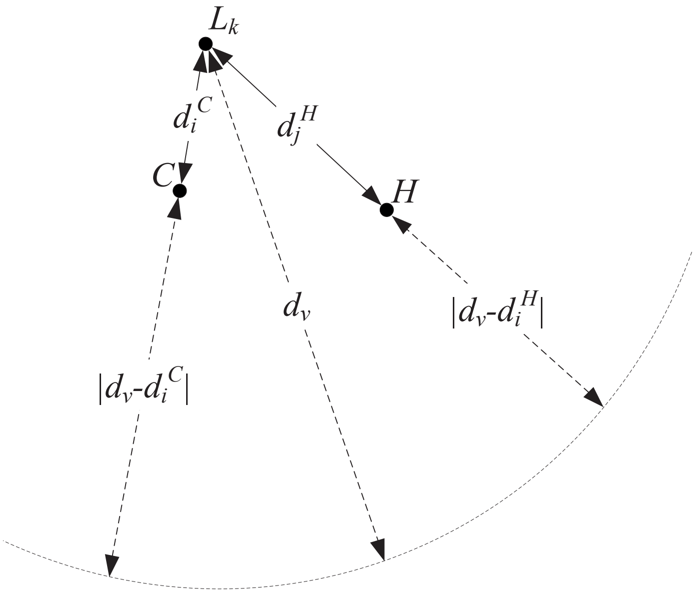


Fig. 3. Key locations and definitions in a simulated scene

If the images captured at C and H are correctly matched to generate a feature matching pair $p_k = \left(f_i^C, f_j^H\right)$, we can use the relation between $f_{i,\sigma}^C$, and $f_{j,\sigma}^H$ to describe the relation between d_i^C and d_j^H. It is apparent that the distance difference $\left|d_v - d_i^C\right|$ at C is larger than $\left|d_v - d_j^H\right|$ at H:

$$\left|d_v - d_i^C\right| > \left|d_v - d_j^H\right|. \tag{2}$$

If $p_k = \left(f_i^C, f_j^H\right)$ is considered as a correct matching pair, there must be $f_{i,\sigma}^C > f_{j,\sigma}^H$. Therefore, we can acquire the following equivalence relation:

$$\left|d_v - d_i^C\right| > \left|d_v - d_j^H\right| \Leftrightarrow d_i^C < d_j^H \Leftrightarrow f_{i,\sigma}^C > f_{j,\sigma}^H. \tag{3}$$

In general, the visual homing methods take only two panoramic images captured at the current location and the home location as input, so the robot cannot directly determine the explicit location distribution of the landmarks and the capturing locations. After acquiring the equivalence relation (3), we can partially handle the distribution problem by using the easy-to-obtain σ values of the SIFT features. To make a more general definition, we denote $\Delta\sigma$ as the scale difference between the two SIFT features of the matching pair:

$$\Delta\sigma = f_{i,\sigma}^C - f_{j,\sigma}^H. \tag{4}$$

The qualitative relation between scale and distance can thus be formulated as follows:

$$\begin{cases} d_i^C < d_j^H, & if \ \ \Delta\sigma > 0 \\ d_i^C > d_j^H, & if \ \ \Delta\sigma < 0. \end{cases} \tag{5}$$

In fact, the proposed MLBH approach is not only reliant on SIFT, the features with associated scale information can also be used (SURF in particular). Although SURF accelerates the computation time, it is not as sensitive to scale variations as SIFT [31]. Since the scale variations of the landmarks are very crucial in our approach, we tested different types of the features, and the experimental results proved that the optimal performance of MLBH can be obtained using SIFT.

3.3. Machine learning-based homing. In this section, we will introduce MLBH in detail. Figure 4 shows the basic model of MLBH. The origin of the world coordinate system is set to the C. μ is the perpendicular dissector of the segment $\overline{CH}$, and it divides the world coordinate plane into two parts, denoted as α and β, respectively. F_1 and F_2 are two arbitrary points separately lies on α and β. We wish to compute the home vector $\boldsymbol{h}$ pointing from C to H.

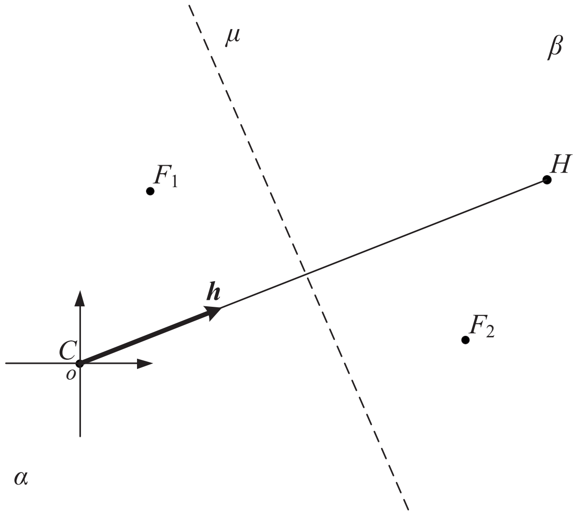


Fig. 4. Basic model of MLBH

In the current image, we define the group of the SIFT features with $\Delta\sigma > 0$ as G_p, and the group of the SIFT features with $\Delta\sigma > 0$ as G_n. According to the geometric theory, the distance from F_1 to C must be shorter than the one from F_1 to H, the situation of F_2 is as opposed to F_1. Obviously, F_1 belongs to G_p, F_2 belongs to G_n. Thus, based on the qualitative relation in Section 3.2, it can be inferred that all the features in G_p must be on α, all the features in G_n must be on β. μ is not only the perpendicular dissector of the segment $\overline{CH}$, but the boundary of G_p and G_n.

Aiming at deriving the expression of μ, we employ the linear SVM classifier to categorize the SIFT features and generate the classification decision boundary. SVM is a type of supervised learning model that analyzes data utilized for classification or regression, it can provide robust classification result even if some abnormal data exist [32]. The standard for obtaining the best hyperplane (i.e. decision boundary) is to separate the two classes with the maximum margin, [33]. Hence, compared with other linear classifiers (e.g. neural network), SVM pursues the most reasonable classification instead of only minimizing errors, so that SVM can effectively neglect the abnormal data (i.e. inaccurate landmarks) and avoid over-fitting.

In MLBH, we consider all the detected SIFT features in the current image as our training data, which take the coordinates originated from C as the feature vector. In the meantime, as mentioned above, we have already labeled the data with –1 or 1 according to the computed $\Delta\sigma$ value. To train a robust linear SVM classifier, we adopt the k-fold cross validation method: The initial samples are randomly partitioned into k sub-samples of the same size, then one of the sub-samples are defined as the validation data for testing, and the remaining k-1 sub-samples are defined as the training data. The cross-validation process is carried out by k times, with each sub-samples used only once as the validation data. In our implementation, we set k to 10, which is a commonly used value.

After the classifier has been trained, we can readily derive the expression of μ by SVM as follows:

$$\mu_x \cdot x + \mu_y \cdot y + b = 0 \tag{6}$$

where μ_x, μ_y and b are three constants. Once the training data is determined, the above parameters will remain the same. Nevertheless, for calculating the home vector, we need to standardize the parameters: If the calculated μ_y value is negative, all the three parameters are turned into their opposite number forms. Since the right side of Equation (6) is equal to 0, the transformed expression is identical to before.

For the purpose of calculating $\boldsymbol{h}$, a crucial conclusion is drawn that $\boldsymbol{h}$ has a constant geometric position relation with the decision boundary: Regardless of the position of the decision boundary in the world coordinate system, $\boldsymbol{h}$ is always perpendicular to it, with the direction pointing from G_p to G_n.

The unit-length home vector $\boldsymbol{h}$ can thus be quantitatively computed. If G_p is above the decision boundary:

$$\boldsymbol{h} = \left(-\frac{\mu_x}{\sqrt{\mu_x^2 + \mu_y^2}}, -\frac{\mu_y}{\sqrt{\mu_x^2 + \mu_y^2}}\right) \tag{7}$$

if G_p is below the decision boundary:

$$\boldsymbol{h} = \left(\frac{\mu_x}{\sqrt{\mu_x^2 + \mu_y^2}}, \frac{\mu_y}{\sqrt{\mu_x^2 + \mu_y^2}}\right). \tag{8}$$

3.4. Procedure of MLBH. The procedure of the mobile robot's homing process based on MLBH is depicted in Fig. 5. The details are described as follows:

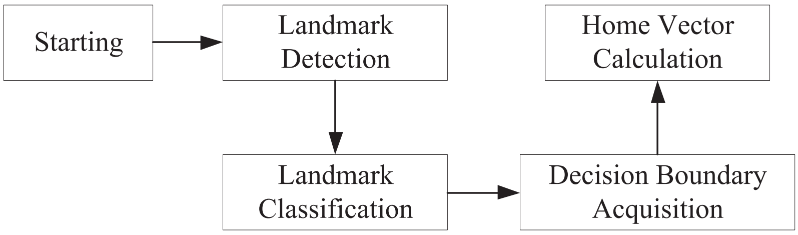


Fig. 5. Procedure of mobile robot's homing scheme based on MLBH

- landmark detection: we detect the SIFT features from the current image and the goal image, then match the features to form the natural landmarks.
- landmark classification: all the landmarks in the current image are categorized into G_p and G_n according to the scale information.
- decision boundary acquisition: SVM is employed to acquire the expression of the decision boundary μ after G_p and G_n are determined.
- home vector calculation: according to the distribution of G_p, G_n and μ, the home vector is yielded based on Equation (7) or (8).

4. Experiments

In this paper, experiments based on both panoramic image databases and actual mobile robot were conducted to evaluate the homing performance of MLBH.

4.1. Panoramic image databases. The panoramic image databases used in this paper were collected at Bielefeld University [34]. These databases provide standard panoramic image information, and many literatures have adopted them for visual homing research. The images were all captured by the panoramic imaging system in different scenes. The environment of the panoramic image databases is a 4.8 m × 2.7 m indoor space, and the images were collected uniformly spaced 30 cm apart. The resolution of these images is 752 × 564.

Six image versions of the panoramic image databases were used for experiments, each containing $17 \times 10 = 170$ panoramic images. The samples of the above versions are shown in Fig. 6, and the brief descriptions of them are presented as follows:

1) Original: the default situation of the room. The doors and the windows were closed, and the overhead fluorescent lights were on.
2) Chairs: three additional chairs were randomly positioned in the room.

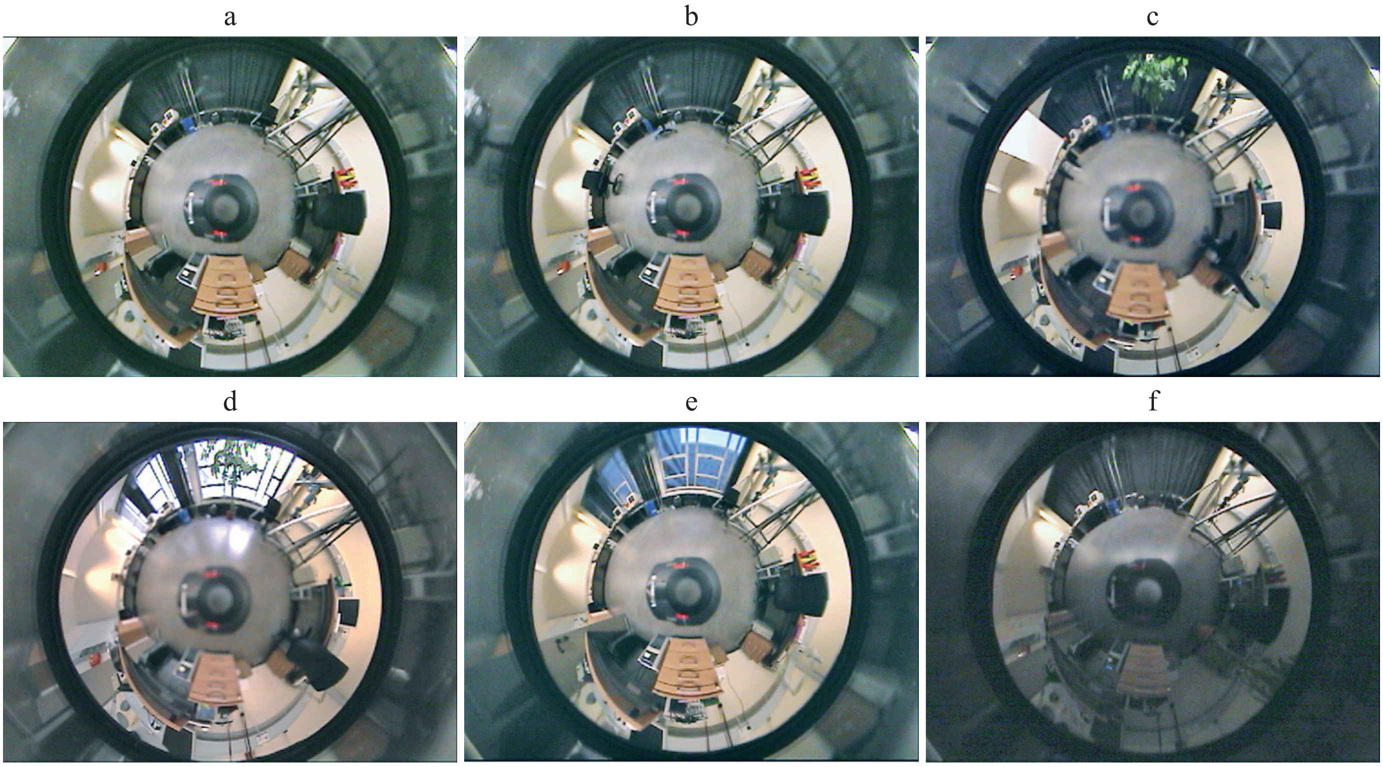


Fig. 6. Samples of panoramic image databases. (a–f) six image versions: original, chairs, screen, day, twilight, winlit

3) Screen: a projection screen was positioned in the center of the room.
4) Day: the curtains were open in full daylight.
5) Twilight: the curtains and doors were open in the twilight.
6) Winlit: The light bars near the door were switched off.

4.2. Parameter settings for SIFT. In general, the number of the training data has a positive correlation with the precision of the final decision boundary, so we appropriately modify some parameters of the original SIFT implementation to maximize the feature production while maintaining the overall precision. The specific modifications are stated as follows:

1) The number of the levels and octaves in scale space is increased from 6 to 10 so that more transformed images can be generated to provide richer scale information.
2) The distance ratio between the nearest neighbor and the second-nearest neighbor is increased from 0.6 to 0.8 so that more feature matching pairs will be produced.

4.3. Performance metrics. In this paper, three widely used metrics are adopted to characterize the performance of visual homing algorithms quantitatively.

The first performance metric is average angular error (AAE). We define the actual homing angular $\theta(H, C)$ and the ideal homing angular $\theta(H, C)$ as follows:

$$\begin{cases} \theta\left(H, C\right) = \mathrm{atan2}\left(h_y, h_x\right) \\ \theta\left(H, C\right) = \mathrm{atan2}\left(y_H - y_C, x_H - x_C\right) \end{cases} \tag{9}$$

where $\left(h_x, h_y\right)$ is 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 then be computed by:

$$AE(H, C) = \left|\theta(H, C) - \theta(H, C)\right| \tag{10}$$

where $AE(H, H) = 0$. For $m \times n$ capture positions in the environment, we employ the average angular error $AAE(H)$ to calculate the average homing performance of a single version:

$$AEE(H) = \frac{1}{mn}\sum_{q=1}^{mn} AE\left(H, C_q\right) \tag{11}$$

where C_q represents the qth current location. According to the distribution of the capture positions in each image version, m is set to 17 and n is set to 10. To measure the performance for all the image versions, we define the overall average angular error $OAAE(V)$, which calculates the total AAE for the entire image databases:

$$OAAE(H) = \sum_{i=1}^{v} AEE_i(H) \tag{12}$$

where v is the total number of the image versions, i is the corresponding index. Low AAE indicates that the robot can move towards the home location with good rapidity and high precision.

The second performance metric is return ratio (RR). It expresses the proportion of all possible current locations which can successfully reach the destination. For all the capture positions in the experiments, a dummy robot is created to perform a simulated movement according to the home vector. The robot moves at a step of $r_{li} = 0.5$, which is the ratio between the step length of the robot and the sampling interval of the images. $r_{bf}(H, C)$ is defined as a binary function to reveal the results of homing process from C to H with the range of [0, 1]. If the robot continually computes the home vectors and moves by r_{li} until H is reached, the value of $r_{bf}(H, C)$ is set to 1. If the robot fails to arrive at H when it travels more than half the circumference of the whole capture grids, the value of $r_{bf}(H, C)$ is set to 0. RR is computed to represent the successful homing rate of all possible C locations to a known H location by:

$$RR(H) = \frac{1}{mn}\sum_{q=1}^{mn} R_{bf}\left(H, C_q\right) \tag{13}$$

where $r_{bf}(H, H) = 1$. High RR indicates that the robot can return from more possible locations to the destination.

The third performance metric is total distance error (TDE). If the current homing process is determined to be successful, we measure the total distance of the actual movement of the robot $d(H, C)$ and compute the ideal minimum distance $d(H, C)$. Then the total distance error $TDE(H)$ can be calculated by:

$$TDE(H) = \left|d(H, C) - d(H, C)\right|. \tag{14}$$

To obtain the overall results of all the image versions, we define the overall total distance error $OTDE(V)$, which computes the overall TDE for the entire image databases:

$$OTDE(V) = \sum_{i=1}^{v} TDE_i(H). \tag{15}$$

TDE is the performance metric applicable to the actual trials in the real scene. By measuring the actual trajectory of the robot, we can assess the deviation of the robot's motion. Low TDE indicates that the robot can reach the destination with high efficiency and good robustness.

4.4. Experiments on image databases. Experiments were carried out based on the image versions introduced in Section 4.1.

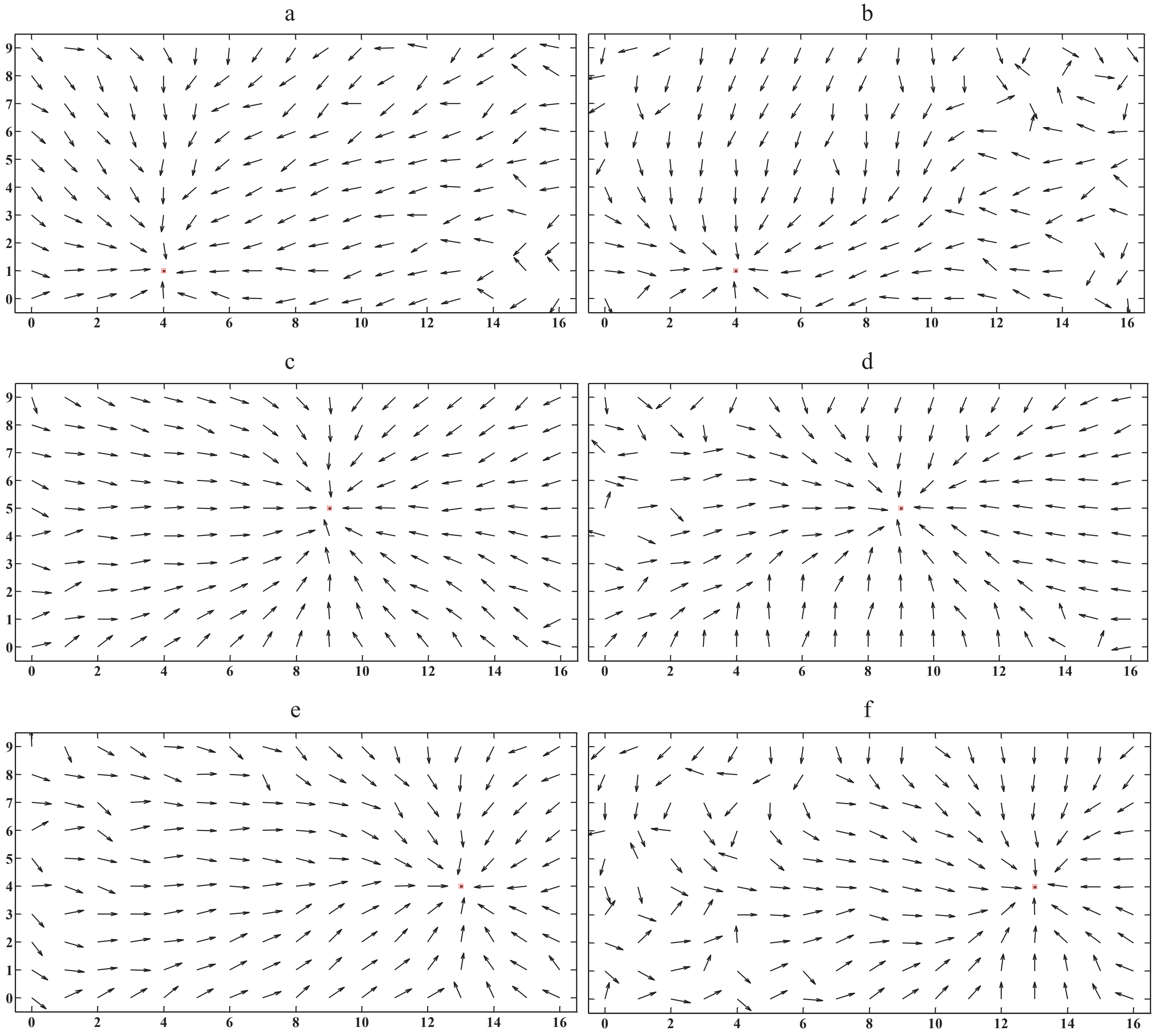


Fig. 7. Home vector fields of MLBH and SL-ALV at (4.1), (9.5) and (13.4) under original version. The whole plots simulates the robot's mobile space containing 17×10 capture positions. The red squares represent the pre-set home location. The arrows represent the calculated home directions pointing from all possible current locations to the pre-set destination. (a, c, e) The home vector fields of MLBH. (b, d, f) The home vectors fields of SL-ALV

These versions were selected mainly for two purposes. First, by the image version of original, chairs and screen, the homing performance was evaluated with different scenes. Second, by the image version of day, twilight and winlit, the homing performance was evaluated with different illumination.

We employed the SIFT features with modified parameters stated in Section 4.2, then compared and analyzed the experimental results of MLBH and SL-ALV. Since different two images had different degrees of similarity, the number of the SVM training data (i.e. natural landmarks) changed each time. According to our experimental results, about 100 $\sim$ 600 natural landmarks were typically detected, the number increased as the distance from the current location to the destination decreased.

In order to assess the overall performance of the homing approaches, we arbitrarily selected a capture position as the home location while the remaining 169 positions were set as all possible current locations, and calculated the home vector from each possible current location to the home location. We call the overall visualization results the home vector field. Multiple home vector fields are shown in Fig. 7. (4,1), (9,5) and (13,4) under the original version were separately selected as destinations, which are discretely distributed in the experiment scene. Figure 8 plots a representation of the *AE* grids corresponding

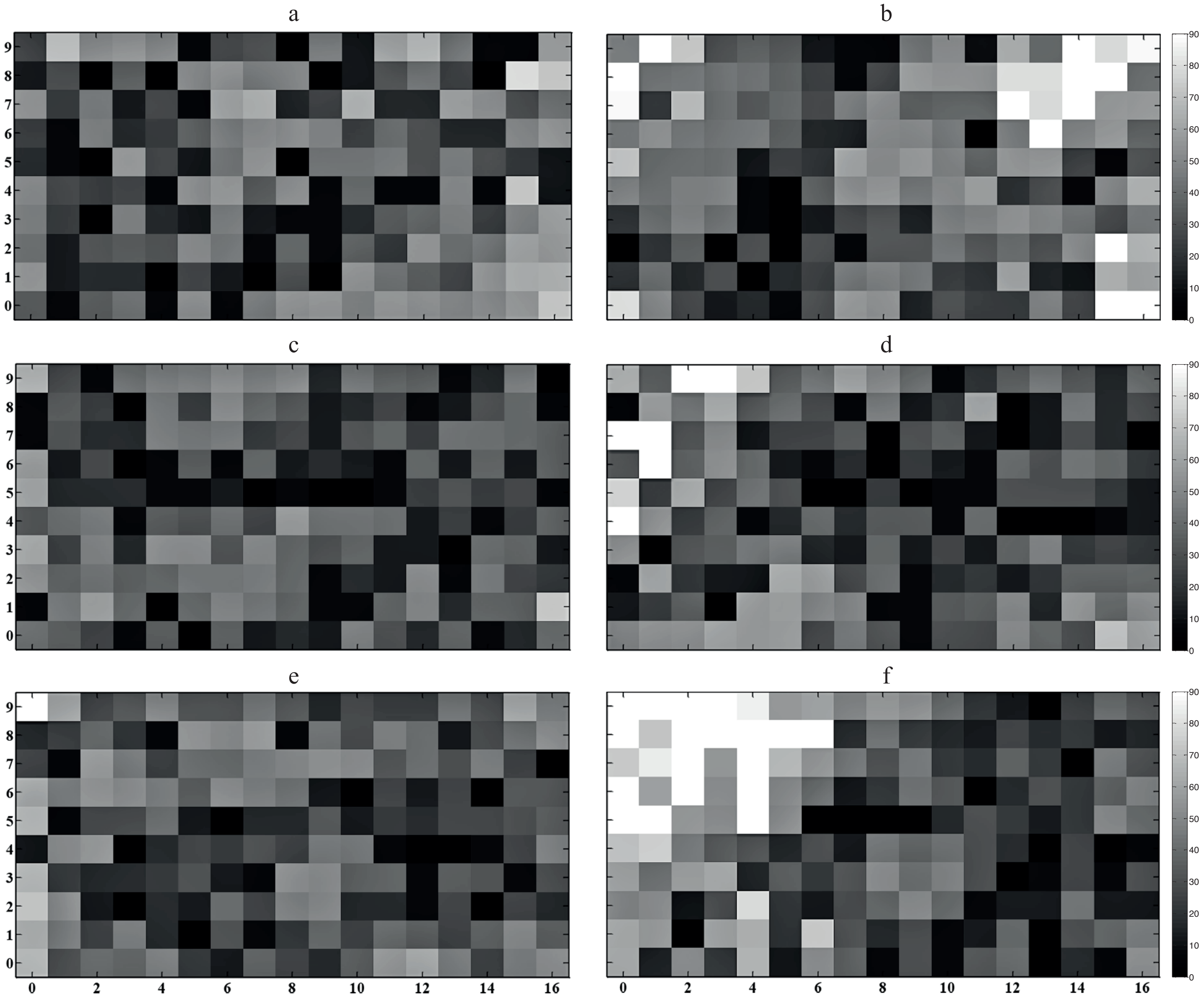


Fig. 8. *AE* grid plots of MLBH and SL-ALV at (4.3), (9.5) and (13.4) under original version. The color change of the unit block from black to white shows a gradually increasing process of *AE* from 0° to 90°. (a, c, e) *AE* grid plots of MLBH. (b, d, f) *AE* grid plots of SL-ALV

to the home vector fields, overall darker plots represent better effects. The calculation results of *AAE* are shown in Table 1. It can be seen intuitively from the above experimental results that MLBH exhibits better precision than SL-ALV at most of the current locations. Besides, MLBH has low *AE* values compared with SL-ALV, especially for the locations far from the destination. With respect to the selected destinations close to the edge of the experiment scene, since they were far from most possible current locations, the *AAE* values of both algorithms increase. However, the increase in *AAE* using MLBH is significantly smaller.

Table 1
AAE results for the two methods

Method	*AAE*(°)		
	(4,1)	(9,5)	(13,4)
MLBH	16.63	10.59	14.09
SL-ALV	31.28	19.22	31.02

Figure 9 shows the *RR* results of MLBH and SL-ALV. The home locations are still set to (4,1) (9,5) and (13,4). For each home location, we select all of the six image versions to perform the experiments. We can conclude that the *RR* results of MLBH is significantly higher than SL-ALV. The number of the locations which can successfully reach the destination using MLBH is approximately 24.33% more than using SL-ALV. The selection of the home location has a lower impact on the *RR* results of MLBH. In particular, when a remote home location is selected, MLBH has the capability to

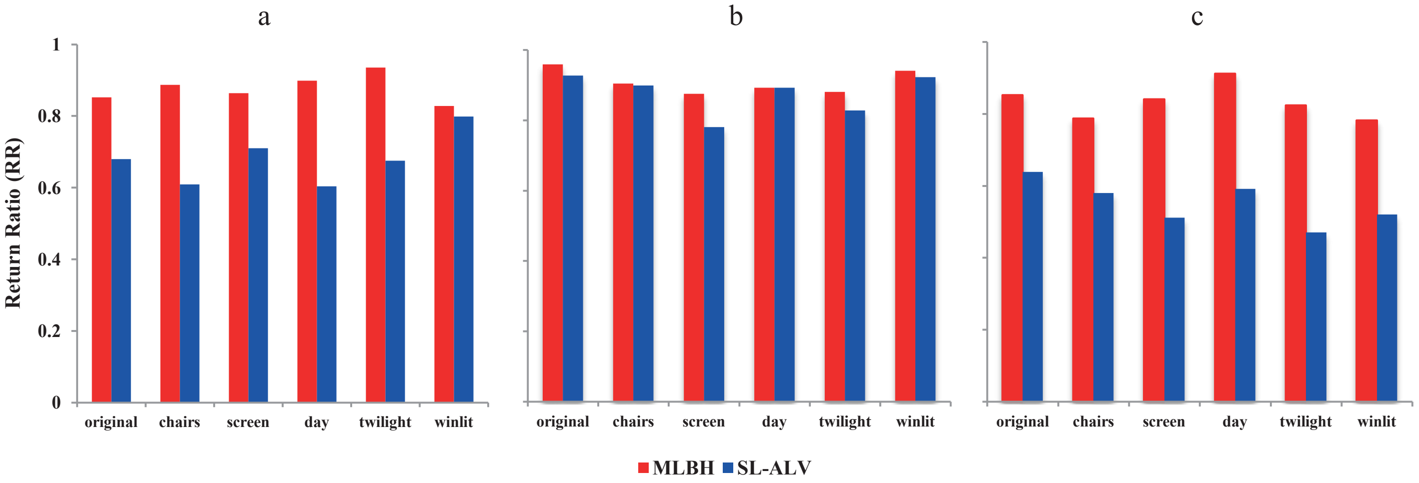


Fig. 9. *RR* results of MLBH and SL-ALV. (a–c) The *RR* results of six versions at (4.1), (9.5) and (13.4). The blue and red histograms are respectively denoted as the *RR* results of MLBH and SL-ALV

maintain robustness, while the *RR* results of SL-ALV dropped sharply.

4.5. Experiments on actual mobile robot. To further evaluate the practical performance of MLBH, related experiments were performed on our omni-directional mobile robot. The selected mobile robot uses mecanum wheels that can guide the robot to move in any direction without changing the robot's orientation. The panoramic vision system was mounted on the robot. Figure 10a and Fig. 10b respectively show the selected mobile robot and the layouts in the environment.

The detailed experimental process is as follows:

Step 1: when the robot is stationary, it captures the panoramic images at its current location.

Step 2: the robot travels 30 cm in a straight line according to the homing direction calculated by MLBH/ SL-ALV, and then pauses.

Step 3: if the following two cases happen, jump to Step 5.

Case 1: the robot arrives within 30 cm of the home location.

Case 2: the robot travels more than 25 steps (at this time, the robot has moved half the circumference of the trial area) or out of the trial area.

Step 4: continue to perform Step 1.

Step 5: if Case 1 happens, we declare the homing process successful and record the robot's total number of steps *N*. If Case 2 happens, we declare the homing process failed.

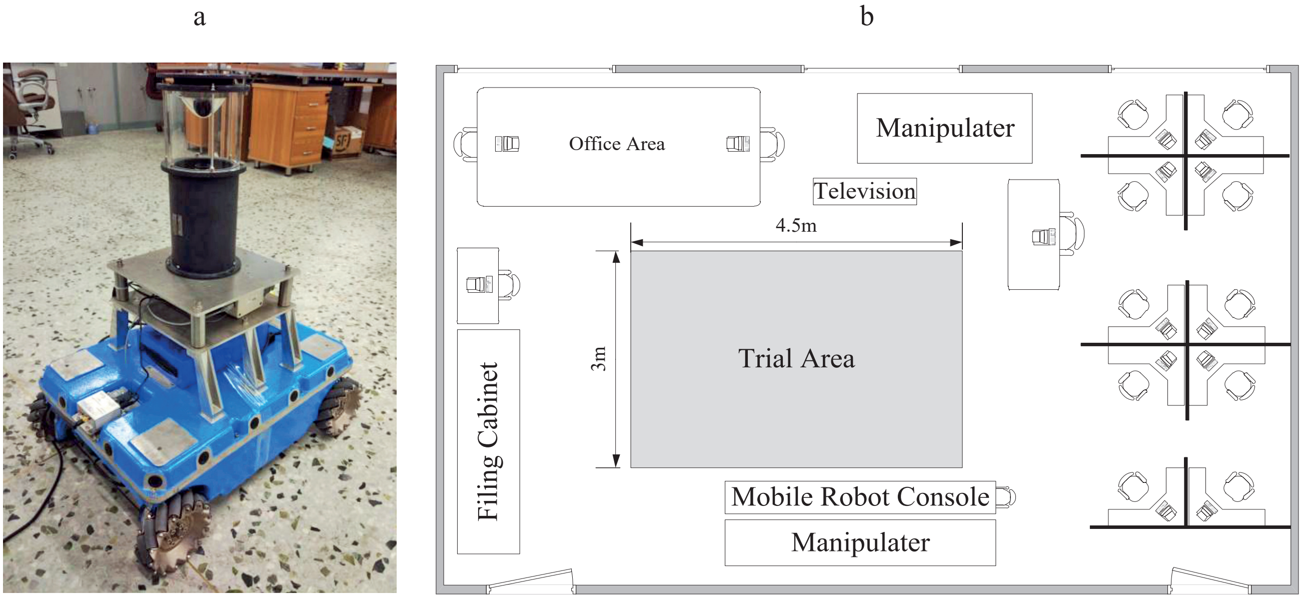


Fig. 10. Robot platform and trial environment. a) Onmi-directional mobile robot; b) layouts of the environment

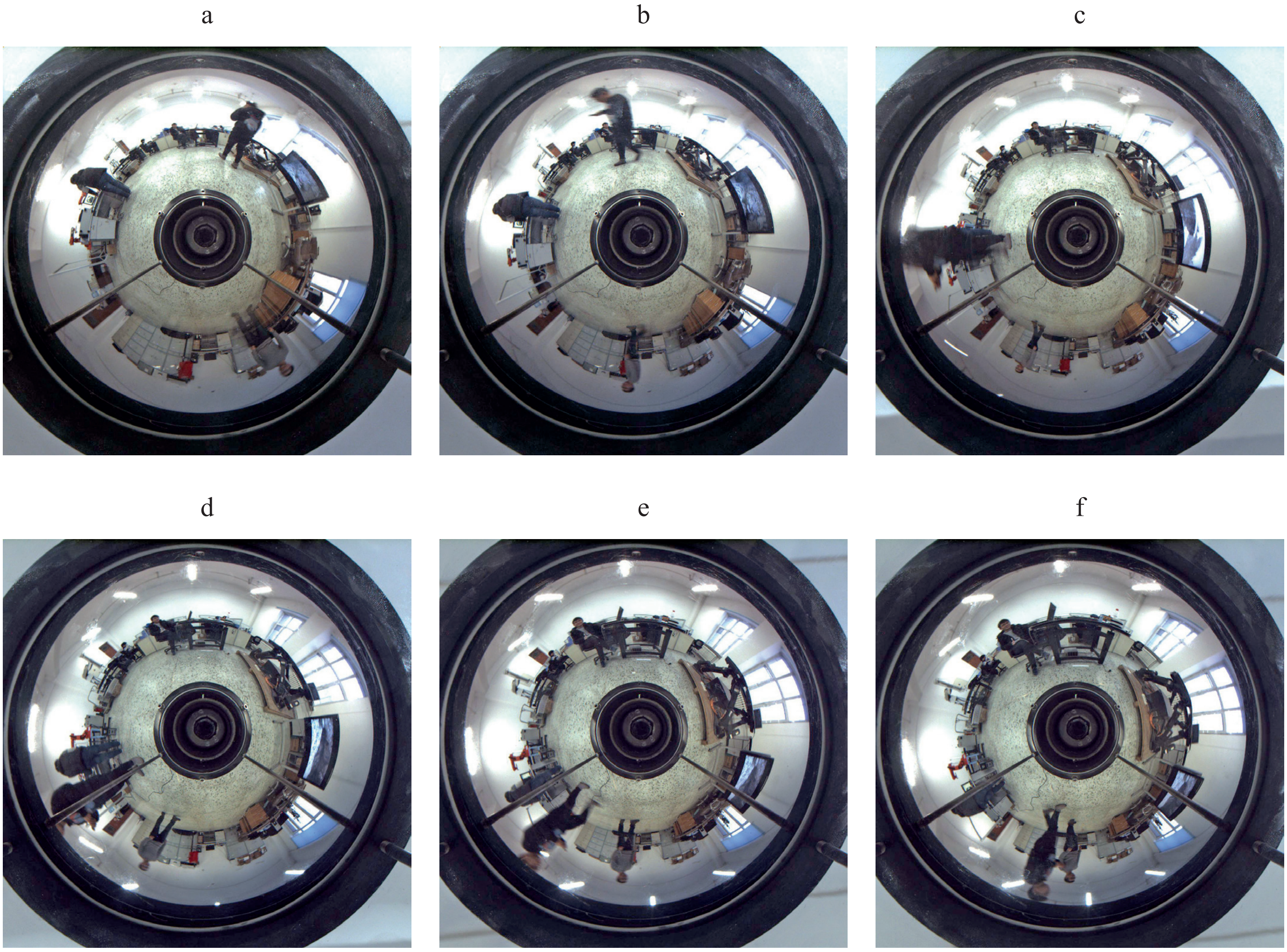


Fig. 11. Image sequence samples collected by the robot during the homing process

In order to create a dynamic environment, we took three measures during the robot's homing process. First, the people moved freely around the robot; Second, the positions of some objects (such as chairs, easy-to-carry experimental equipments, boxes, etc.) were changed arbitrarily; Third, the lights were turned on/off randomly. Figure 11 shows the image sequence samples collected by the robot during its homing process.

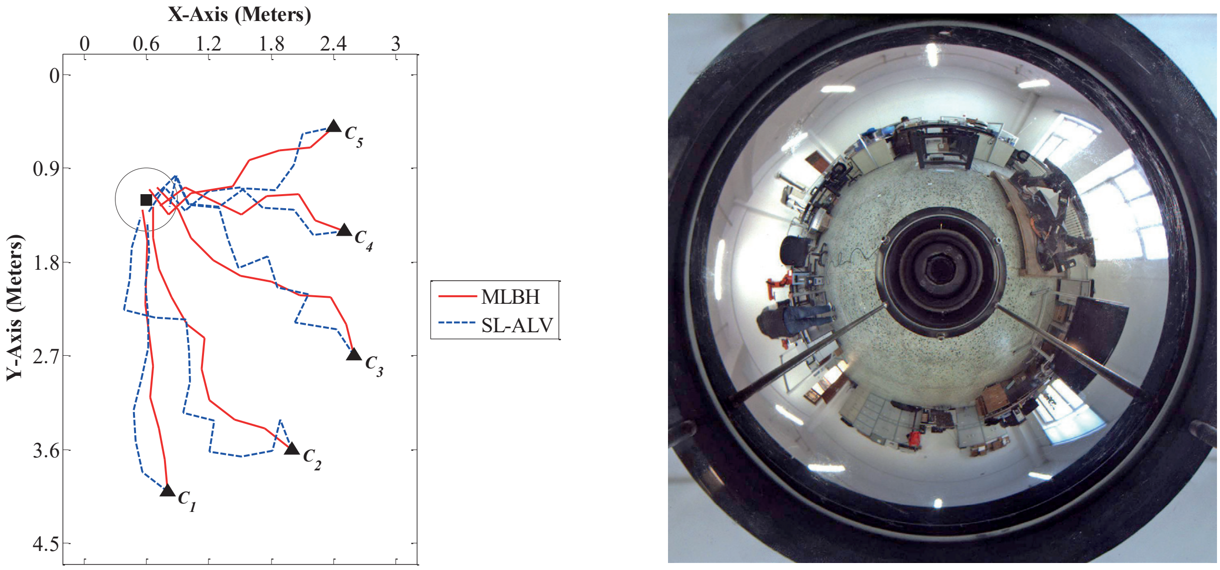
To obtain a general conclusion, we selected five discrete home locations, with five different starting locations for each home location. The starting locations were arbitrarily selected and distributed as evenly as possible throughout the trial area for tests. Figures 12–16 show the actual homing trajectories of the robot guided by two methods along with a table of associated performance metrics. According to the experimental results on actual mobile robot, about 150 $\sim$ 800 natural landmarks were typically detected.

Note that the actual homing trajectories in Figs. 12–16 have verified the good effect of MLBH. Compared with SL-ALV, MLBH has significantly lower angular and distance errors, as well as higher successful homing rates. A total performance metric evaluation of all the homing trials of MLBH and SL-ALV are shown in Table 2. The results are consistent with the simulation results based on the panoramic image databases.

Table 2
Total performance metric statistics of MLBH and SL-ALV

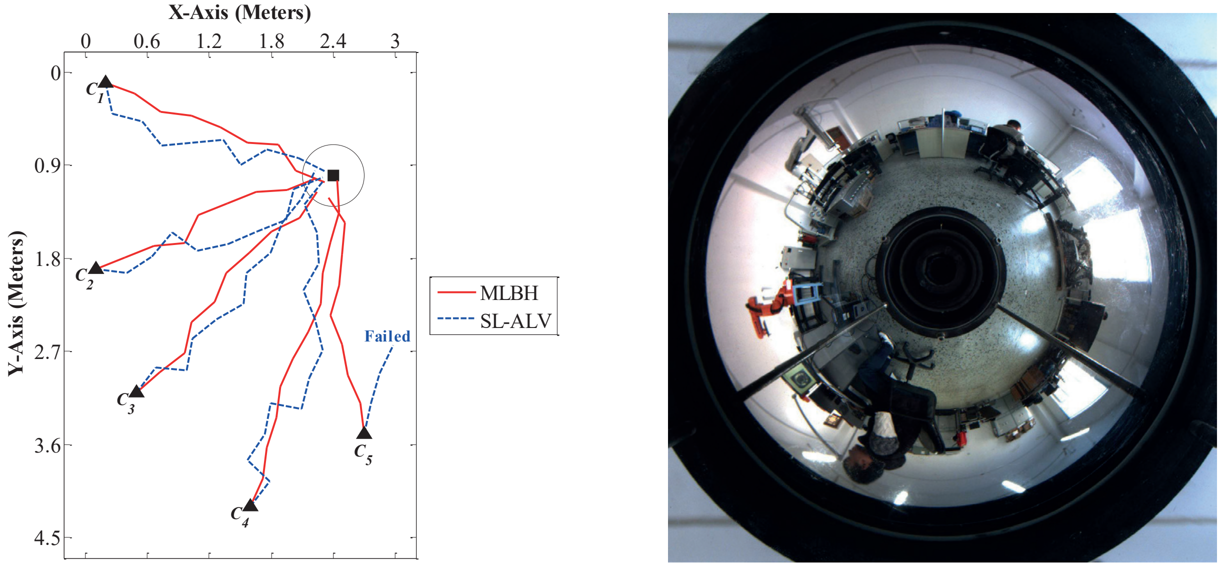
Metrics	MLBH	SL-ALV
Trial times	25	25
Successful homing times	25	23
OAAE **(°)**[*]	66.80	117.91
OTDE **(m)**[*]	7.16	17.06
Total homing steps[*]	172	205

[*] Only the data when both of MLBH and SL-ALV have successfully returned to the home area are taken into account.



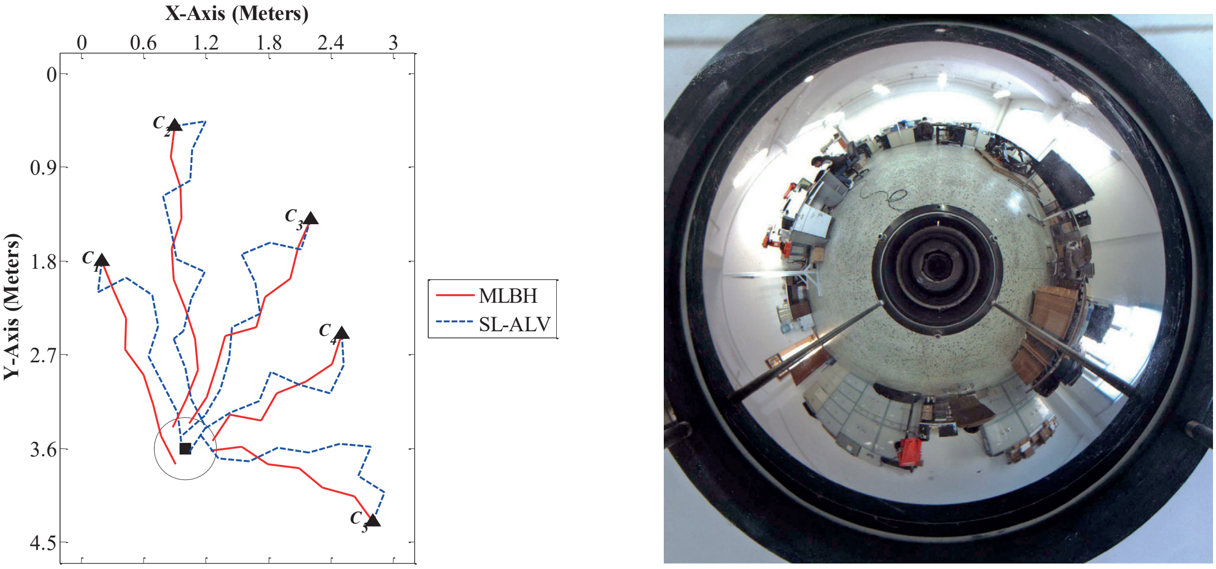
Metrics	C_1		C_2		C_3		C_4		C_5	
	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AAE **(°)**	5.04	11.29	18.68	39.63	23.94	29.45	22.45	21.04	18.54	22.39
TDE **(m)**	0.19	0.19	0.52	1.72	0.50	1.10	0.48	0.78	0.47	0.77
N	9	9	10	14	9	11	7	8	7	8

Fig. 12. Actual homing trial 1. The home area is marked as a circle centered on a square block. The five starting locations C_1, C_2, C_3, C_4 and C_5 are marked as triangular blocks. Red solid lines represent the homing trajectories calculated by MLBH. Blue dashed lines represent the homing trajectories calculated by SL-ALV. Top left: the robot's homing trajectories from five different current locations. Top right: the panoramic image captured at the home location. Below: associated performance metrics



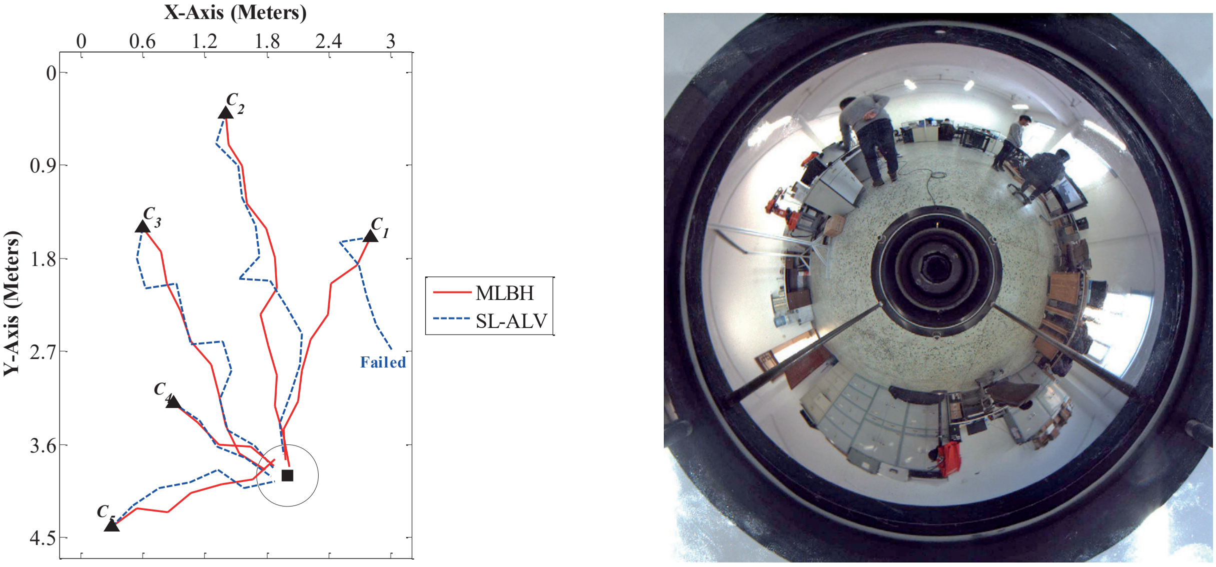
Metrics	C_1		C_2		C_3		C_4		C_5	
	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
AAE **(°)**	11.33	23.72	10.03	21.54	10.59	20.62	8.75	25.82	15.43	—
TDE **(m)**	0.32	0.32	0.23	0.53	0.17	0.47	0.30	0.90	0.18	—
N	8	8	8	9	9	10	11	13	8	—

Fig. 13. Actual homing trial 2



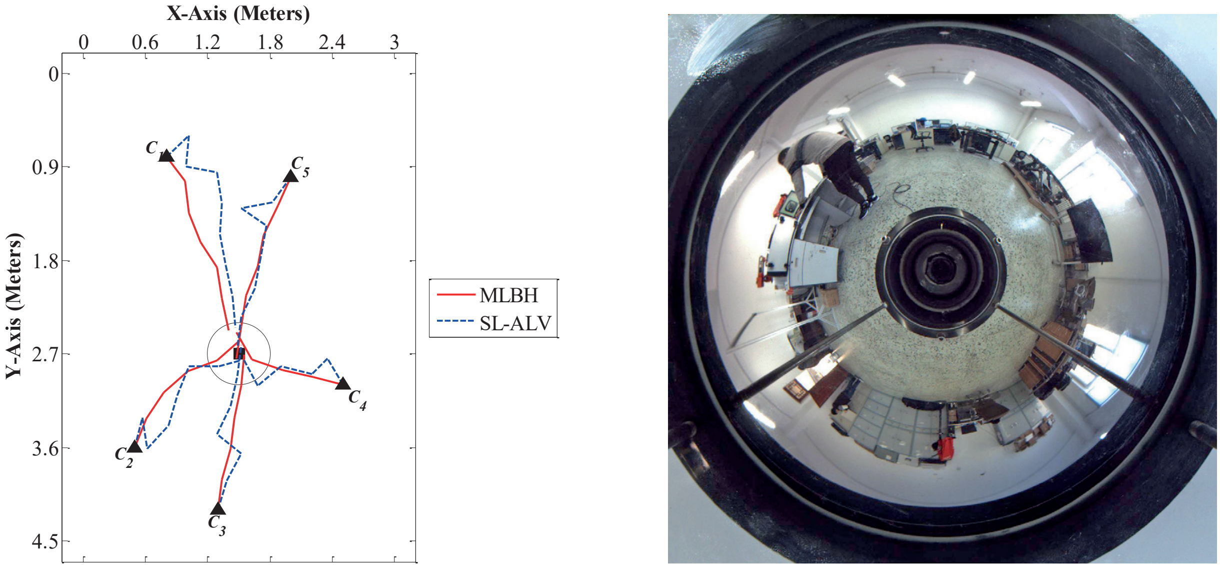
Metrics	C_1		C_2		C_3		C_4		C_5	
	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
***AAE*(°)**	14.79	25.89	12.52	29.42	17.92	25.40	17.45	23.52	13.99	28.55
***TDE*(m)**	0.43	0.73	0.20	1.40	0.19	0.79	0.24	0.54	0.17	1.07
N	7	8	10	14	8	10	6	7	6	9

Fig. 14. Actual homing trial 3



Metrics	C_1		C_2		C_3		C_4		C_5	
	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV
Success	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
***AAE*(°)**	12.26	—	11.36	22.25	12.72	26.06	18.67	18.39	17.43	17.83
***TDE*(m)**	0.26	—	0.35	0.65	0.22	0.82	0.20	0.20	0.37	0.37
N	8	—	12	13	9	11	4	4	6	6

Fig. 15. Actual homing trial 4



Metrics	C_1		C_2		C_3		C_4		C_5	
	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV	MLBH	SL-ALV
Success	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AAE **(°)**	8.73	26.70	11.77	30.24	7.16	14.29	8.68	21.29	6.25	20.16
TDE **(m)**	0.08	0.68	0.45	1.05	0.29	0.59	0.46	0.76	0.33	0.63
N	6	8	5	7	5	6	4	5	6	7

Fig. 16. Actual homing trial 5

5. Conclusions

In this paper, we propose a novel and robust visual homing approach called MLBH. The main advantages of MLBH can be summarized as follows.

1. The model of MLBH is simple and easy to implement, the calculation process of the home vector is straightforward without exhaustive search or vector transformation.
2. MLBH can overcome the constraint on the spatial distribution assumption of landmarks, so that the landmarks can be arbitrarily selected as needed.
3. MLBH can reduce the impact of the inaccurate landmarks on performance, so it is suitable for both static and dynamic environments.

Experimental results revealed that MLBH had significantly low *AAE* and high *RR*, and 100% homing success rate was obtained according to our actual experiment. However, same as most homing approaches, MLBH also requires the robot to keep its orientation consistent while capturing images. Although the omni-directional mobile robot with mecanum wheels helps us solve the problem, the application of MLBH is still limited with respect to a general mobile robot. In the future, we will pay more attention to the image registration problem in order to overcome the constraint of the homing algorithms on the robot's orientation.

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

References

[1] 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).

[2] M. Hoy, A.S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey", *Robotica* 33(3), 463–497(2015).

[3] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Pseudo-bacterial potential field based path planner for autonomous mobile robot Navigation", *Int. J. Adv. Robot. Syst.* 12(81), 1–14 (2015).

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

[5] N. Ohnishi and A. Imiy, "Appearance-based navigation and homing for autonomous mobile robot", *Image Vis. Comput.* 31(6), 511–532 (2013).

[6] 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).

[7] A. Kim and R.M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment", *Int. J. Robot. Res.* 34(4–5), 457–475 (2015).

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

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

[10] B.A. Cartwright and T.S. Collett, "Landmark learning in bees", *J. Comp. Physiol.* A151, 421-543 (1983).

[11] A. Denuelle and M.V. Srinivasan, "Bio-inspired visual guidance: From insect homing to UAS navigation" *IEEE Int. Conf. Robot. Biomim*, pp. 326–332 (2015).

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

[13] M.O Franz, B. Schölkopf, H.A. Mallot, and H.H Bülthoff, "Where did I take that snapshot? Scene-based homing by image matching", *Biol. Cybern.* 79(3), 191–202 (1998).

[14] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner, "A mobile robot employing insect strategies for navigation", *Robot. Auton. Syst.* 30(1), 39–64 (2000).

[15] T. Murray and J. Zeil, "Quantifying navigational information: The catchment volumes of panoramic snapshots in outdoor scenes", *PloS one*, 12(10), e0187226 (2017).

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

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

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

[19] R. Möller, "A SIMD implementation of the MinWarping method for local visual homing", Bielefeld University, Germany, 2016.

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

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

[22] R. Möller, "Column distance measures and their effect on illumination tolerance in MinWarping", Bielefeld University, Germany, 2016.

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

[25] 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).

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

[28] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "The role of homing in visual topological navigation", in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 567–572 (2012).

[29] 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).

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

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

[32] P. Zarychta, P. Badura, and E. Pietka, "Comparative analysis of selected classifiers in posterior cruciate ligaments computer aided diagnosis", *Bull. Pol. Ac.: Tech.* 65(1), 63–70 (2017).

[33] J. Nayak, B. Naik, and H.S. Behera, "A comprehensive survey on support vector machine in data mining tasks: Applications & challenges", *Int. J. Database Theory Appl.* 8(1), 169–186 (2015).

[34] Panoramic Image Databases. Available online: http://www.ti.uni-bielefeld.de/html/research/avardy/index.html (accessed on 15 April 2017).