Chapter 2
Angle-Based Navigation Using the 1D Trifocal Tensor

2.1 Introduction

In the present chapter, we study how visual information can be used to endow a robot moving on the ground plane with the ability to autonomously navigate toward a specific position in an environment. Completing this task requires the robot to possess information representative of the target location, and therefore, the task may be understood as the robot *returning* to a previously visited place. Due to this interpretation, this behavior has commonly received the name *homing*. This is one of the fundamental competencies for a mobile robot and has multiple applications across different domains (i.e., in domestic, industrial, or field robotics). In addition, it is an attractive behavior both in single-robot and multirobot scenarios. In the latter case, a team of robots may be required to reach a set of positions in order to provide coverage of an environment, or to create a robotic formation.

Let us now concentrate on the specific problem addressed in the chapter, i.e., robot homing, and review relevant literature. As discussed in the introduction of the monograph, vision sensors have long been used to perform robot navigation [1] and continue to attract interest in the present [2], due to the abundance of information provided by cameras and their relatively low cost. For the task called *visual homing* (i.e., homing carried out using visual information), the solutions that have been proposed in the literature are often inspired by the mechanisms that certain animal species, such as insects, utilize to return to their known home location [3, 4]. A wide variety of methods have been employed to perform this task, e.g., employing information from the angles of image features [5], image distances [6] or frequency components [7], performing image warping [8], or utilizing the scale of image features [9, 10], their distances [11] or their vertical displacement [12].

The method presented in this chapter is firstly motivated by the properties of omnidirectional vision sensors. They offer a wide field of view, which is a very interesting quality for navigation, and provide very precise angular information. In contrast, their radial information is strongly affected by distortion. In addition to this, it is known that the robustness of the performance of vision-based tasks can be improved if,

© Springer International Publishing AG 2017

M. Aranda et al., *Control of Multiple Robots Using Vision Sensors*,
Advances in Industrial Control, DOI 10.1007/978-3-319-57828-6_2

instead of using the image information directly (either through appearance-based measurements or extracted features), the geometric models that relate the views of a scene are employed. Thus, we propose a homing method for a robot moving on the ground plane that makes use of the angles between omnidirectional views extracted by means of the 1D trifocal tensor model. Our approach, which takes advantage of the planar motion constraint through the 1D tensor, employs only the angular visual information provided by omnidirectional images to obtain the relative angles between the view from the current position and a set of previously acquired reference views taken at different locations, any of which can be selected as the home (or goal) position. A stable control law based on the estimated angles is used to guide the robot to the target.

2.1.1 Related Work

Let us next review the literature that is closely related with the method we propose in the chapter. The interesting properties of omnidirectional cameras have been the motivation behind a number of angle-based homing methods, being [13] an early work and [5, 14] more recent contributions. These approaches are purely feature-based: Landmark angles in the omnidirectional images are used to generate a homing vector, which defines the direction of motion toward the goal. An alternative that is known to increase the robustness in the presence of wrong feature matches, which is one of our motivations throughout this work, is to employ the models that express the intrinsic geometric relations between the views of a scene. A number of visual control methods have been presented using two-view models such as the epipolar geometry [15–18] and the homography [19–21].

The trifocal tensor is the model that encapsulates the geometric constraints of three views of a scene [22]. This model has been used for control purposes [23, 24]. In particular, robot navigation on the ground plane lends itself to the use of the 1D trifocal tensor, the matching constraint between three 1D views which minimally parameterizes them [25, 26]. The 1D trifocal tensor, which can be computed linearly from point correspondences, allows to perform 2D projective reconstruction. In robotics, it has been used for 2D localization tasks [27, 28] and control [29].

When compared with other approaches in the literature of visual navigation, ours is neither a pure topological map-based nor image memory-based approach, but it shares elements with both kinds of techniques. As the representation of the robot's environment necessary to perform navigation, we use a set of omnidirectional images. These serve us not only to build a connectivity graph of the environment, as in [30, 31], but also to store further information, which fits the *view graph* concept of [32]. We use the extracted geometric information relating the views to find connections between all the available reference images. Thus, in contrast with [30, 31], the graph we construct is not appearance-based, but as close to complete as possible. In addition to the graph, we also store the geometric information needed to perform homing between any two connected nodes in it using our proposed angle-based control law.

Some works in the literature use an image memory in a visual path-following scheme [5, 20, 33]. We also employ a memory of stored images, which in our case serves as a representation of the environment that provides the references necessary to compute the geometric information that permits navigation. However, in contrast with these methods, our approach allows direct homing between two distant positions without having the need to travel along a rigid visual route of intermediate images. This long-range ability of our technique also differentiates it from local homing approaches, which are typically short-range [3, 4, 6, 7, 9, 15]. The reason is that these techniques require the robot to be in a position where the visual information overlaps with that available at the target position (i.e., the robot is in the so-called *catchment area*). This limitation does not exist in our method.

2.1.2 Contributions Relative to the Literature

An important contribution of the presented approach which sets it apart from the other visual homing approaches is the use of the 1D trifocal tensor to extract the geometric information. In addition to being particularly appropriate for the characteristics of omnidirectional cameras and planar motion, as discussed above, this tensor proves to be a strong constraint which allows the extraction of angular information in a remarkably robust way. The reasons for this are the presence of three views in the model and the possibility to perform triangle consistency checks to validate the estimations. This increases the robustness with respect to purely feature-based methods and approaches employing two-view geometric constraints. In addition, the angle-based control law we present has strong stability properties, further contributing to the robustness of the overall system.

We also present as contribution a method to compute the relative angles between all the available omnidirectional views. This is achieved through the use of the computation of the 1D trifocal tensor, a new angular disambiguation procedure, and a novel approach for the indirect computation of angles. This method is what endows our technique with long-range direct homing capabilities.

The chapter is organized as follows: Sect. 2.2 presents the procedure for the computation of all the angular information needed for the homing task. In Sect. 2.3, the designed control strategy and its implementation details are described, and an analysis of its stability is presented. Section 2.4 provides an account of the results of the simulations and experiments conducted with real images. Finally, in Sect. 2.5, a discussion of the method is presented. Some nomenclature and conventions used throughout the chapter are illustrated in Fig. 2.1.

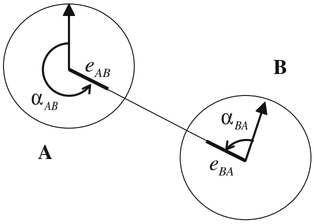

Fig. 2.1 Nomenclature and conventions used throughout the chapter. e_{AB} is the epipole of view B in view A. α_{AB} is the angle or direction of that epipole, i.e., its angular polar coordinate in view A. The reference axis for the measurement of the angles is given by the robot's direction of motion. The angles are measured counterclockwise between $-\pi$ and π

2.2 Computation of the Reference Set Angles

From the set of omnidirectional images we use to represent the environment, we can define an undirected graph $\mathscr{G}(\mathscr{V}, \mathscr{E})$. The nodes or vertices which make up the set $\mathscr{V}$ are the locations on the plane where each of the images was acquired. A link exists between two given nodes when the relative angles between their corresponding associated views are known, and $\mathscr{E}$ is the set of these links. Therefore, we can think of this graph as a topological map (i.e., a representation of an environment as a group of places and a group of connections between them). The adjacency matrix ($\mathbf{A}$) of $\mathscr{G}$ is defined such that $\mathbf{A}(i, j) = 1$ when a link in $\mathscr{E}$ exists between nodes i and j in $\mathscr{V}$. Otherwise, $\mathbf{A}(i, j) = 0$. In addition to the graph description, which expresses the connectivity of the nodes in our system, we must also store the relative angles between the views. We do so using a data structure $\mathbf{M}$ such that $\mathbf{M}(i, j) = \alpha_{ij}$ is the angle of the epipole of view j in view i.

Relevant features are extracted and matched between pairs of images on the reference set, and the resulting point correspondences are stored. We then start an estimation procedure that operates as follows:

- A set of four images (which we can call A, B, C, and D) taken in two groups of three (e.g., A–B–C and B–C–D) are processed in each step. For each trio of images, we obtain three-view point correspondences by taking the common two-view matches between them. From a minimum number of seven (or five, if the points are calibrated) matches between the three views in each group, we can calculate two 1D trifocal tensors, and we can eventually obtain the angles of the epipoles in each of the views of the four-image set (as will be described in the following sections). This way, the four associated graph nodes become adjacent.
- We run through the complete set of reference images calculating trifocal tensors and estimating the angles between the views. Whenever there is more than one estimation of a certain angle available, we simply choose the result that was obtained from the largest set of point matches. A possible alternative would be to compute a weighted average. After this stage is completed, we usually end up with an incomplete graph.

- We fill the graph's adjacency matrix using the indirect estimation procedure that will be described in Sect. 2.2.2. This procedure is aimed at computing the unknown angles between the views from the already known values. For every pair of nodes i, j such that $\mathbf{A}(i, j) = 0$, we search for two other nodes k, l such that $\mathbf{A}(i, k)$, $\mathbf{A}(i, l)$, $\mathbf{A}(j, k)$, $\mathbf{A}(j, l)$, and $\mathbf{A}(k, l)$ all equal 1. When these conditions hold, we are able to compute the angles between i and j and consequently make $\mathbf{A}(i, j) = 1$ and calculate the value $\mathbf{M}(i, j)$.

The typical way in which the graph's adjacency matrix becomes filled is the following: When several locations are adjacent physically, the relative angles between them can be computed directly thanks to the large amount of common visual information between the views, and the graph nodes corresponding to these locations are also adjacent. Thus, initially, we have the whole set of images divided in groups of connected locations (with adjacent associated graph nodes). If two of these groups are adjacent physically, it is very likely that they have at least two nodes in common. When this is the case, we can connect all the elements in the two groups using the indirect angle computation procedure, and all the nodes in the two groups become adjacent. Following this procedure, the adjacency matrix gets gradually filled until every pair of nodes are adjacent (i.e., the graph is complete) or it is not possible to find new connections between the nodes. Next, we describe the different steps of the process through which the relative angles between the views are obtained.

2.2.1 *Angles from the 1D Trifocal Tensor*

The trifocal tensor is the mathematical entity that encapsulates all the geometric relations between three views that are independent of scene structure. In particular, the 1D trifocal tensor relates three 1D views on a plane and presents interesting properties; namely, it can be estimated linearly from a minimum of seven three-view point matches (or five, if the calibration of the cameras is known [34]), and 2D projective reconstruction can be obtained from it.

2.2.1.1 Foundations: 1D Trifocal Tensor Computation and Epipole Extraction

The projections of a given point in three 1D views (which we will refer to as A, B, and C) on a plane are related by the following trilinear constraint [26]:

$$\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} T_{ijk} u_i^A u_j^B u_k^C = 0, \tag{2.1}$$

where T_{ijk} are the elements of the *1D trifocal tensor*, $\mathbf{T} \in \mathbb{R}^{2\times2\times2}$, between the views, and $\mathbf{u^A}$, $\mathbf{u^B}$, and $\mathbf{u^C}$ are the homogeneous coordinates of the projections of the point in each view. $\mathbf{T}$ is defined up to a scale factor and therefore can be calculated, in the uncalibrated case, from a minimum set of seven point correspondences across the views. If the points are calibrated, only five correspondences are required.

The process we follow to estimate $\mathbf{T}$ starts by detecting relevant features in three omnidirectional images, e.g., by means of the SIFT keypoint extractor [35], and finding matches between them. The angles (α) of the matched image points, measured counterclockwise from the vertical axis, are converted to a 1D projective formulation, with the corresponding homogeneous 1D coordinates being $(\sin\alpha, \cos\alpha)^T$. In this mapping, the distinction between angles differing by π is lost. The geometry behind this 1D imaging formulation is illustrated in Fig. 2.2.

Each of the point matches in 1D projective coordinates gives rise to an equation of the form of (2.1). If more than the minimum number of correspondences is available, we find a least squares solution to the resulting system of linear equations through singular value decomposition [22, 28]. In this process, a robust estimation method (RANSAC) is employed in order to reject wrong matches.

After $\mathbf{T}$ has been estimated, we extract the epipoles from it using a procedure taken from [27, 36] that we briefly describe next. A 1D homography is a mapping between projected points in two lines (two of the 1D views, in our case) induced by another line. From the coefficients of the trifocal tensor, we can directly extract what are known as the *intrinsic homographies*. For example, the two intrinsic homographies from A to B, $\mathbf{K_{AB}}$ and $\mathbf{L_{AB}}$, are obtained by substituting the lines defined by $\mathbf{u^C} = (1, 0)^T$ and $\mathbf{u^C} = (0, 1)^T$ in (2.1), yielding:

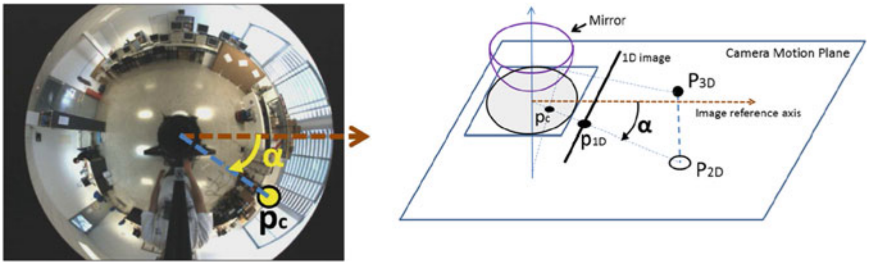

Fig. 2.2 Illustration of the 1D imaging-based formulation we employ. *Left* a catadioptric omnidirectional camera whose image plane is parallel to the ground plane and whose motion is planar, projects a scene point $\mathbf{P_{3D}}$ in an image point $\mathbf{p_c}$. From the angular image measurement corresponding to this point, α, it is possible to define a representation in terms of an equivalent 1D perspective camera, which is illustrated on the *right-hand* side of the figure. This camera projects in its 1D image the point $\mathbf{P_{2D}}$, which results from considering the two-dimensional coordinates of the 3D point $\mathbf{P_{3D}}$ in the real camera motion plane. Considering that the 1D camera is such that its focal length is equal to 1, then the homogeneous calibrated coordinates of the projected point are: $\mathbf{p_{1D}} = (\sin\alpha, \cos\alpha)^T = (\tan\alpha, 1)^T$ (Note the angular measurements are measured clockwise in the depicted example. Still, the angles can be equivalently measured counterclockwise)

$$\mathbf{K_{AB}} = \begin{bmatrix} -T_{211} & -T_{221} \\ T_{111} & T_{121} \end{bmatrix}, \ \mathbf{L_{AB}} = \begin{bmatrix} -T_{212} & -T_{222} \\ T_{112} & T_{122} \end{bmatrix}. \qquad (2.2)$$

Now, $\mathbf{H_A} = \mathbf{K_{AB}}\mathbf{L_{AB}^{-1}}$ is a homography from A to itself; by definition, the epipoles are the only points that are mapped to themselves by such a homography, i.e., $\mathbf{e_{AB}} = \mathbf{H_A}\mathbf{e_{AB}}$ and $\mathbf{e_{AC}} = \mathbf{H_A}\mathbf{e_{AC}}$. Therefore, we can calculate them as the eigenvectors of matrix $\mathbf{H_A}$. It is important to note, though, that with this method we do not know which of the other two views (B or C) each of the two recovered epipoles corresponds to. By mapping this pair of epipoles to the other views through the intrinsic homographies, we finally obtain the six epipoles of the set of three 1D views.

2.2.1.2 Ambiguity Resolution

There are three ambiguities that need to be resolved in order to determine the correct values of the angles of the 2D epipoles from the values of the epipoles extracted using the 1D trifocal tensor.

Firstly, as mentioned in Sect. 2.2.1.1, an epipole in a given view recovered from the 1D trifocal tensor may be assigned to any of the two other views. This results in two possible solutions in the assignment of the set of six epipoles between the three views. As shown in [26, 34], both solutions give completely self-consistent 2D projective reconstructions, regardless of the number of point matches between the views. This fundamental ambiguity in the 2D reconstruction from three 1D views can only be resolved through the use of a fourth view, as noted in [27]. We propose a new method to resolve the ambiguity that operates in the following way: having a group of four views (which we can call A, B, C and D), we calculate two different trifocal tensors between them; for instance, the tensor relating A, B and C, and the tensor between B, C, and D. Since the epipoles between B and C must be identical in the two estimations, by detecting the common (or, in a real situation, the closest) epipoles in these two views we can disambiguate the assignment of the complete set of epipoles.

The origin of the two other ambiguities lies in the fact that in the mapping of 2D image points to 1D projective coordinates, the distinction between bearings differing by π is lost. The angle of a recovered 1D epipole $(e_1, e_2)^T$ is obtained as $\arctan(e_1/e_2)$ in 2D. As a consequence, from the 1D epipole, we can extract two different angles in a 2D view, separated by π radians. There are, therefore, four possible solutions for the values of the epipoles between two given views A and B, which may be interpreted as emerging from two combined reconstruction ambiguities, namely an ambiguity in the direction of the translation vector from view A to view B, which accounts for the difference between solutions (a) and (c) in Fig. 2.3, and an ambiguity of π radians in the orientation of view B, illustrated, for example, by solutions (a) and (b) in the same figure.

This double ambiguity for a set of two views might be resolved through point reconstruction, but instead we propose a simple method employing only the angles of matched image points. The method takes advantage of the properties of the optical

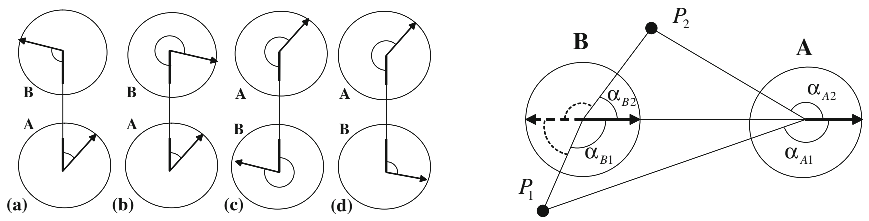

Fig. 2.3 Four possible 2D reconstructions from the epipoles between two views extracted from the 1D trifocal tensor (*left*). The relations between the angles of the projections of matched points (e.g., P_1 and P_2) in two aligned views can be used to resolve the 2D reconstruction ambiguities (*right*)

Algorithm 2.1 Disambiguation of the relative angles between views A, B, C and B, C, D from the 1D epipoles

1. Let us consider the 1D epipoles in inhomogeneous coordinates in view B: e_{B1}, e_{B2} extracted from the tensor computed between A, B and C, and $e^{'}_{B1}$, $e^{'}_{B2}$ extracted from the tensor computed between B, C and D. Find $i \in \{1, 2\}, j \in \{1, 2\}$ such that $\min\{|\arctan(e_{Bi}) - \arctan(e^{'}_{Bj})|, |\arctan(e_{Bi}) - \arctan(e^{'}_{Bj}) - \pi|\}$ is minimum. Then compute $\alpha_{BC} = (\arctan(e_{Bi}) + \arctan(e^{'}_{Bj}))/2$, and assign the angles of all the other epipoles accordingly. The angles have an ambiguity of π.
2. For each pair of images (let us take A and B as example):
   a. Define $\alpha^s_{AB} = \{\alpha_{AB} \vee \alpha_{AB} + \pi\}$ and $\alpha^s_{BA} = \{\alpha_{BA} \vee \alpha_{BA} + \pi\}$ so that both α^s_{AB} and $\alpha^s_{BA} \in (0, \pi]$
   b. Rotate all the m points matched between A and B (α_{Ak} and α_{Bk}, measured counterclockwise):
      For $k = 1, \ldots, m$, $\alpha_{Akr} = \alpha_{Ak} - \alpha^s_{AB}$, $\alpha_{Bkr} = \alpha_{Bk} - \alpha^s_{BA}$
   c. Test 1: Relative orientation of the two aligned images. For $k = 1, \ldots, m$, $nr(k) = sign(\alpha_{Akr} \cdot \alpha_{Bkr}) \cdot min(|\sin\alpha_{Akr}|, |\sin\alpha_{Bkr}|)$. If $\sum_{k=1}^{m} nr(k) > 0$, then $test_1 = 0$, otherwise $test_1 = 1$
   d. Test 2: Sign of translation from A to B. If $test_1 = 1$, then for $k = 1, \ldots, m$, $\alpha_{Bkr} = \alpha_{Bkr} + \pi$.
      For $k = 1, \ldots, m$, if $sign(\alpha_{Akr} \cdot \alpha_{Bkr}) = 1$ then $st(k) = sign(|\alpha_{Akr}| - |\alpha_{Bkr}|) \cdot (|\alpha_{Akr}| - |\alpha_{Bkr}|)^2$,
      otherwise $st(k) = 0$. If $\sum_{k=1}^{m} st(k) > 0$, then $test_2 = 0$, otherwise $test_2 = 1$
   e. Obtain from Table 2.1 the unambiguous angles α_{AB} and α_{BA}
3. Check the joint coherence of the disambiguations of the pairs of views taken in groups of three, by verifying in Table 2.2 that the obtained three-view reconstructions are valid.

Table 2.1 Disambiguation of the angles of the epipoles in two views

$test_1$	$test_2$	α_{AB}	α_{BA}	Case in Fig. 2.3
1	1	α^s_{AB}	α^s_{BA}	(a)
0	1	α^s_{AB}	$\alpha^s_{BA} + \pi$	(b)
1	0	$\alpha^s_{AB} + \pi$	$\alpha^s_{BA} + \pi$	(c)
0	0	$\alpha^s_{AB} + \pi$	α^s_{BA}	(d)

Table 2.2 Possible reconstructions of camera locations from epipoles in three views

$P_B =$	$[R_B \vert t_B]$	$[R_B \vert t_B]$	$[-R_B \vert -t_B]$	$[-R_B \vert -t_B]$
$P_C =$	$[R_C \vert t_C]$	$[-R_C \vert -t_C]$	$[R_C \vert t_C]$	$[-R_C \vert -t_C]$
$P_B =$	$[R_B \vert -t_B]$	$[R_B \vert -t_B]$	$[-R_B \vert t_B]$	$[-R_B \vert t_B]$
$P_C =$	$[R_C \vert -t_C]$	$[-R_C \vert t_C]$	$[R_C \vert -t_C]$	$[-R_C \vert t_C]$

flow between two omnidirectional images when they are aligned. To do so, it exploits the relative angular differences between the projections of a scene point in the two images. This is illustrated in Fig. 2.3. The complete procedure for the computation of the relative angles between views from the 1D trifocal tensor, resolving all the existing ambiguities, is provided in Algorithm 2.1.

Next, we determine what the possible reconstructions are for a trio of 1D views in the plane, starting from the knowledge of the 1D epipoles between them. As we have already seen, a 1D epipole can represent two different angles in the plane separated by π radians, which gives four possible reconstructions of the 2D relative geometry between two views. With three views, we have six epipoles, and the combination of all the possible values of the angles gives $2^6 = 64$ possible reconstructions of the geometry of the views. However, by considering geometric consistency constraints between the three views, it can be shown that only $2^3 = 8$ of these possibilities is actually valid, jointly coherent reconstructions.

When we convert the points in three omnidirectional images with a common image plane into 1D coordinates, the situation is equivalent to having three 1D cameras in a common 2D space. We can express the relative locations of these cameras with a rotation matrix $\mathbf{R} \in \mathbb{R}^{2\times 2}$ and a translation vector $\mathbf{t} \in \mathbb{R}^{2\times 1}$. By considering one of the cameras ($\mathbf{P_A}$) as the fixed origin of the reference system, we have the following projection matrices: $\mathbf{P_A} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{P_B} = [\mathbf{R_B}|\mathbf{t_B}]$ and $\mathbf{P_C} = [\mathbf{R_C}|\mathbf{t_C}]$. The rotation matrices are as follows:

$$\mathbf{R_B} = \begin{bmatrix} \cos\phi^s_B & \sin\phi^s_B \\ -\sin\phi^s_B & \cos\phi^s_B \end{bmatrix}, \; \mathbf{R_C} = \begin{bmatrix} \cos\phi^s_C & \sin\phi^s_C \\ -\sin\phi^s_C & \cos\phi^s_C \end{bmatrix}, \tag{2.3}$$

where we can express: $\phi^s_B = \pi - \alpha^s_{AB} + \alpha^s_{BA}$ and $\phi^s_C = \pi - \alpha^s_{AC} + \alpha^s_{CA}$. The superscript s alludes to angles extracted directly from the 1D epipoles arbitrarily (i.e., without having been disambiguated). The translation vectors can also be worked out:

$$\mathbf{t_B} = \begin{bmatrix} -e_{AB}\cos\phi_B^s - \sin\phi_B^s \\ e_{AB}\sin\phi_B^s - \cos\phi_B^s \end{bmatrix}, \ \mathbf{t_C} = \begin{bmatrix} -e_{AC}\cos\phi_C^s - \sin\phi_C^s \\ e_{AC}\cos\phi_C^s - \cos\phi_C^s \end{bmatrix}, \tag{2.4}$$

where e_{AB} and e_{AC} are the epipoles in inhomogeneous coordinates (i.e., they equal the value of the tangent of the associated angle). The vectors $\mathbf{t_B}$ and $\mathbf{t_C}$ are defined up to scale. Actually, there is only one scale in the three-view reconstruction, since a relation exists between the magnitudes of $\mathbf{t_B}$ and $\mathbf{t_C}$ through the triangle formed by the views:

$$\frac{||\mathbf{t_C}||}{||\mathbf{t_B}||} = \left|\frac{\sin(\alpha_{BA}^s - \alpha_{BC}^s)}{\sin(\alpha_{CA}^s - \alpha_{CB}^s)}\right|. \tag{2.5}$$

The eight possible reconstructions, up to scale, from the epipoles between three views are shown in Table 2.2.

Thus, after using the proposed two-view disambiguation method, we check whether the views taken in groups of three give jointly coherent reconstructions. In addition, we also check that the triangle formed by the locations of the three views is consistent (i.e., that the sum of its angles is sufficiently close to π). By doing so, the angles between every trio of views are estimated robustly.

2.2.2 Complete Solution of Four-View Sets

In practice, it is usually not possible to find matches across all the images. Next, we propose a method to compute all the angular information using the matches between sets of adjacent or close images. A geometric setting of the type shown in Fig. 2.4, where two triangles are known between the locations of four views, comes up in our method every time we estimate two trifocal tensors from a four-view set. This

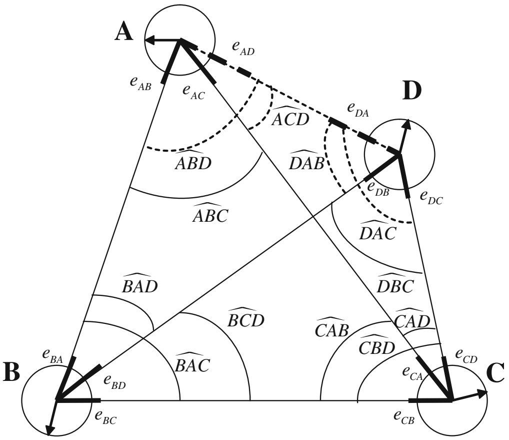

Fig. 2.4 Geometric setting with four views and two known triangles

section describes the method employed to calculate the two unknown angles in this configuration.

We use the notation $\widehat{ABC}$ to allude to the values (>0) of the angles in a triangle. Without loss of generality, we can formulate the problem in the following terms: all the angles from every view to the others in the set are known except the angles of the epipoles between views A and D. Therefore, all the angles in the four triangles formed by the set of four views are known, except the ones including both vertices A and D (represented with dashed lines in Fig. 2.4). Our objective is to calculate the angles α_{AD} and α_{DA} of the epipoles e_{AD} and e_{DA}, which can be directly obtained from the knowledge of the angles of the triangles at those vertices. We start by applying the law of sines on the set of four triangles (ABC, ABD, ACD, and BCD), which finally yields the following expression:

$$\frac{\sin \widehat{ABD}}{\sin \widehat{ACD}} = K_A, \tag{2.6}$$

where K_A is a known value given by:

$$K_A = \frac{\sin \widehat{CBD} \cdot \sin \widehat{BAD}}{\sin \widehat{BCD} \cdot \sin \widehat{CAD}}. \tag{2.7}$$

Using the intrinsic relationship between the three angles at vertex A and applying trigonometric identities, we can calculate the individual values of the angles in (2.6). We must, however, take into account the fact that the location of A with respect to the other three vertices changes the geometry of the set and, consequently, the relation between the angles at the aforementioned vertex. Therefore, we need to divide the plane into seven regions, as shown in Fig. 2.5, to account for these differences. It turns out that the expression that gives the angle $\widehat{ABD}$ has the same form in all cases (i.e., for all regions), but the signs of two of its terms, denoted as $sign_1$ and $sign_2$, are dependent on the region where A lies:

$$\widehat{ABD} = \arctan \frac{sign_1 \cdot K_A \sin(\widehat{ABC})}{1 + sign_2 \cdot K_A \cos(\widehat{ABC})}. \tag{2.8}$$

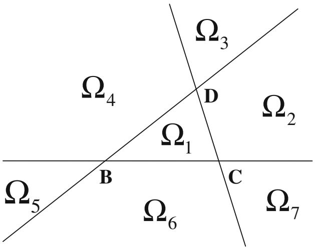

Fig. 2.5 Seven regions where point A can be located

Table 2.3 Values of signs for the different regions in which A may lie

Region of vertex A	Relation between angles at vertex A	$sign_1$	$sign_2$
Ω_1	$\widehat{\mathbf{ACD}} = 2\pi - \widehat{\mathbf{ABD}} - \widehat{\mathbf{ABC}}$	–1	1
Ω_2, Ω_5	$\widehat{\mathbf{ACD}} = \widehat{\mathbf{ABD}} + \widehat{\mathbf{ABC}}$	1	–1
Ω_3, Ω_6	$\widehat{\mathbf{ACD}} = \widehat{\mathbf{ABC}} - \widehat{\mathbf{ABD}}$	1	1
Ω_4, Ω_7	$\widehat{\mathbf{ACD}} = \widehat{\mathbf{ABD}} - \widehat{\mathbf{ABC}}$	–1	–1

We can easily determine the region in which A is located using the known angles of the epipoles in views B and C, and choose the appropriate values of $sign_1$ and $sign_2$ as shown in Table 2.3.

The angle of the epipole of view D in view A is finally obtained as follows:

$$\alpha_{AD} = \begin{cases} \alpha_{AB} + \widehat{ABD}, \text{ if } 0 \leq \alpha_{BA} - \alpha_{BD} < \pi \\ \alpha_{AB} - \widehat{ABD}, \text{ if } 0 > \alpha_{BA} - \alpha_{BD} \geq -\pi \end{cases}. \tag{2.9}$$

The angle of the epipole in view D of view A (α_{DA}) can be calculated through a completely analogous process, simply interchanging the roles of vertices A and D. The results are validated using geometric consistency checks. By employing the procedure we have just presented, we can calculate the two unknown angles and thus obtain the complete set of angles between the four views. In addition, this method is useful for two other purposes within our homing technique:

- In the initial stage, described in the beginning of Sect. 2.2, this method allows to fill in the missing elements in the matrix of epipole angles, corresponding to pairs of views that could not be linked directly due to the impossibility to find a sufficiently large set of three-view matches between them.
- During homing, it enables us to obtain all the angles needed to generate the motion commands employing a minimum number of three views; we only need to compute the trifocal tensor between the current image taken by the robot and two of the reference images, which reduces the cost of the algorithm.

2.3 Homing Strategy

In this section, we describe the strategy designed in order for the mobile robot to perform homing. We assume the robot moves on the ground plane and has *unicycle* kinematics, which model the motion of the most common type of ground robotic platforms, i.e., *differential-drive* robots [37]. The implications of considering this or other *nonholonomic* motion models for robot control will be studied comprehensively in the following chapter of the monograph. The homing method is based solely on the computation of the angles between a series of omnidirectional views of the environment. This group of snapshots consists of the image taken by the robot from its current position and a set of previously acquired reference images, which includes an image obtained at the desired target location. The angles between the views on the reference set have been previously computed and stored, as described in Sect. 2.2. Therefore, only the angles between the current and the reference views must be worked out during homing.

In every step of the robot's motion, the camera takes an omnidirectional image, from which key points are extracted. When sufficient point matches are found between the current and two of the reference images, the 1D trifocal tensor is calculated as detailed in Sect. 2.2.1.1. From the tensor, aided by the knowledge of the angles on the reference set, we can extract the angles between the current and the two other views. Finally, with the method explained in Sect. 2.2.2, all the angles of the epipoles in all the views can be computed.

2.3.1 Control Law

For every reference view $R_i(x_i, z_i, \varphi_i)$ (where x_i, z_i and φ_i define its position and orientation in the ground plane), the difference between the angles of its epipoles with respect to the current and goal views defines an angular sector of size $S_i = |\alpha_{iC} - \alpha_{iG}|$, as illustrated in Fig. 2.6. We use the average value of the angular sizes of these sectors to set the linear velocity at which the robot will move toward the target position:

$$v = k_v \cos\alpha_{CG} \cdot \frac{1}{n}\sum_{i=1}^{n} S_i, \tag{2.10}$$

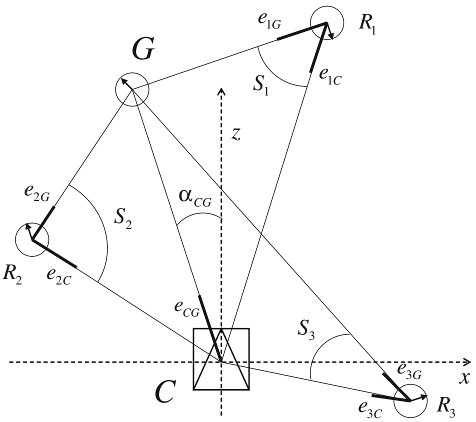

Fig. 2.6 Elements involved and angles employed in the homing strategy. C is the robot's current location, at the coordinate origin. G is the goal location. R_i are reference views. Three of the n views on the reference set are depicted as example

where $k_v > 0$ is a control gain. As the robot moves closer to the goal, the mean size of the angular sectors seen from the reference positions will become smaller; thus, the robot's linear velocity will gradually decrease and eventually become zero when the target is reached. The cosine term in (2.10) ensures that v has the right sign; when the target is behind the robot, $\cos \alpha_{CG}$ will be negative, therefore generating backward motion. In addition, the cosine improves the behavior by gradually reducing the vehicle's translational speed when it is not pointing to the goal.

The direction in which the robot travels is determined by the angle at which the goal position is seen from the current location, i.e., the angle α_{CG} of the epipole e_{CG}. The angular velocity of the control law is given by:

$$\omega = k_\omega(\alpha_{CG} - \alpha_{CG}^d), \tag{2.11}$$

$$\alpha_{CG}^d = \begin{cases} 0 \text{ if } & |\alpha_{CG_0}| \leq \frac{\pi}{2} \\ \pi \text{ if } & |\alpha_{CG_0}| > \frac{\pi}{2} \end{cases},$$

where $k_\omega > 0$ is a control gain, and α_{CG_0} is the value of α_{CG} at the start of the execution. From a minimum number of four reference views, one of which would be the view from the target location, the robot will navigate to the home position. Note that the orientation in which the robot reaches the target position is not controlled, since, by definition, the purpose of the homing task is getting to the goal location. In addition, final orientation correction is less relevant when one uses robots equipped with omnidirectional cameras, since they always provide a full view of the environment regardless of the robot's orientation.

2.3.2 Method Implementation

We present the information required and steps to follow to implement the proposed homing strategy, in Algorithms 2.2 (offline procedure) and 2.3 (online homing).

Algorithm 2.2 Implementation: Offline image capture and epipole angles computation

1. Capture a set of images covering the environment, including all the possible desired destinations for the robot.
2. Extract image descriptors (e.g., SIFT) for each image.
3. Compute matches between descriptors for pairs of images.
4. Compute matches among trios of views from results of the previous step.
5. Compute 1D trifocal tensors and relative angles from the sets of three-view matches (Sect. 2.2.1).
6. Fill up the matrix of epipole angles using the indirect computation method (Sect. 2.2.2).

Algorithm 2.3 Implementation: Online homing

```
I. Designate, from the image set, the goal image, which defines the target location for the robot.
II. While homing not completed do
    1. Capture a new current image $I_c$.
    2. Extract image features of $I_c$, or track previously extracted features.
    3. Match the features with those of any two other images in the image set.
    4. Compute the 1D trifocal tensor and relative angles for this trio of images (Sect. 2.2.1).
    5. Use the matrix of epipole angles to obtain the angles between view $I_c$ and the other views in
       the set (Sect. 2.2.2).
    6. Use the angles to obtain $S_i$ and $\alpha_{CG}$, and compute the control law (2.10), (2.11).
    7. Execute motion command $(v, \omega)$.
```

2.3.3 Stability Analysis

We first define d_i as the distance between reference view i and the goal position (i.e., the length of line segments $\overline{R_i G}$ in Fig. 2.6), and d_{min} as the minimum of such distances. Let us study first the global stability of the system.

Theorem 2.1 *The system under the proposed control law (2.10), (2.11) is globally asymptotically stable if* $k_\omega > k_v \cdot \pi / d_{min}$.

Proof We will use Lyapunov techniques [38] to analyze the stability of the system. We define the following definite positive candidate Lyapunov function:

$$V(\mathbf{x}, t) = \frac{\rho^2}{2} + \frac{(\alpha_{CG} - \alpha_{CG}^d)^2}{2}, \tag{2.12}$$

where ρ is the distance between the current and goal positions, and $\mathbf{x}$ is the state of the system, determined by ρ and α_{CG}. The two state variables we use are a suitable choice, since we are only interested in reaching the goal position, regardless of the final orientation of the robot. As can be seen, both V and $\dot{V}$ are continuous functions.

We note at this moment that the equilibrium in our system occurs at the two following points: $(\rho, \alpha_{CG}) = (0, 0)$ and $(\rho, \alpha_{CG}) = (0, \pi)$, which correspond to the situations where the robot reaches the goal moving forward or backward, respectively. In order to account for the multiple equilibria, in the following we use the global invariant set theorem [39] to prove the asymptotic stability of the system.

What we need to show is that V is radially unbounded and $\dot{V}$ is negative semidefinite over the whole state space. It is straightforward that $V(\mathbf{x})$ is radially unbounded, given that $V(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$. Next, we prove that the derivative $\dot{V}(\mathbf{x})$ is negative definite. For our chosen candidate Lyapunov function, this derivative is as follows:

$$\dot{V} = \rho\,\dot{\rho} + (\alpha_{CG} - \alpha_{CG}^d)\,\dot{\alpha}_{CG}. \tag{2.13}$$

We will suppose that the vehicle on which the control method is to be implemented is a nonholonomic unicycle platform. The dynamics of the system as a function of the input velocities is then given, using the derivatives in polar coordinates with the origin at the goal, by $\dot{\rho} = -v\cos(\alpha_{CG})$ and $\dot{\alpha}_{CG} = -\omega + v\sin(\alpha_{CG})/\rho$. Using the control velocities (2.10), (2.11) we obtain:

$$\begin{aligned}\dot{V} &= -k_v\,\rho\,\cos^2(\alpha_{CG})\,\frac{1}{n}\sum_{i=1}^{n} S_i - k_\omega(\alpha_{CG}-\alpha_{CG}^d)^2 \\ &\quad + (\alpha_{CG}-\alpha_{CG}^d)\cdot\sin\alpha_{CG}\,\frac{k_v}{\rho}\,\cos\alpha_{CG}\cdot\frac{1}{n}\sum_{i=1}^{n} S_i.\end{aligned} \tag{2.14}$$

By definition, $\rho \geq 0$ and $S_i \geq 0$. It is then straightforward to see that the first and the second terms of (2.14) are negative definite. However, the third term can be positive. The interpretation is that for the system to be stable, the convergence speed provided by the angular velocity has to be higher than the convergence speed given by the linear velocity. Otherwise, the angular error is not corrected fast enough and the robot will move following spirals around the goal. Still, the stability can be guaranteed if the control gains are selected properly. From (2.14), we can see that it is guaranteed that $\dot{V} < 0$ if the following inequality holds:

$$|k_\omega\cdot(\alpha_{CG}-\alpha_{CG}^d)| > \left|\sin(\alpha_{CG})\cos(\alpha_{CG})\,\frac{k_v}{\rho}\cdot\frac{1}{n}\sum_{i=1}^{n} S_i\right|. \tag{2.15}$$

This is equivalent to the following condition on the angular velocity gain:

$$k_\omega > \left|\frac{\sin(\alpha_{CG})}{(\alpha_{CG}-\alpha_{CG}^d)}\cos(\alpha_{CG})\cdot k_v\cdot\frac{1}{n}\sum_{i=1}^{n}\frac{S_i}{\rho}\right|. \tag{2.16}$$

We aim to find an upper bound to the right side of (2.16). We start by analyzing the first fraction. Since α_{CG}^d is equal to either 0 or π, and $\sin(\alpha_{CG}) = -\sin(\alpha_{CG}-\pi)$, we have:

$$\left|\frac{\sin(\alpha_{CG})}{(\alpha_{CG}-\alpha_{CG}^d)}\right| = \left|\frac{\sin(\alpha_{CG})}{(\alpha_{CG})}\right| \leq 1, \tag{2.17}$$

as $\sin(\alpha_{CG})/\alpha_{CG}$ is a *sinc* function, whose maximum absolute value occurs at $\alpha_{CG} = 0$ and equals 1. We now look for a bound to the S_i/ρ term in (2.16). The angular sector S_i seen from reference view i has a value lying in the interval $0 \leq S_i \leq \pi$. We will study two subintervals separately:

- $0 \leq S_i \leq \pi/2$. Applying the law of sines on the triangle defined by vertices C, G and R_i in Fig. 2.6, the addend in (2.16) corresponding to reference view i becomes:

$$\frac{S_i}{\rho} = \frac{S_i}{\sin(S_i)} \cdot \frac{\sin(\widehat{CR_iG})}{d_i} \leq \frac{\pi}{2 \cdot d_{min}}. \tag{2.18}$$

The first fraction of the product in (2.18) is a function of S_i whose value equals 1 at $S_i = 0$ and increases monotonically to a value of $\pi/2$ at $S_i = \pi/2$, which is the limit of the interval we are considering. Since the second fraction has an upper bound equal to $1/d_{min}$, the product of the two is upper-bounded by $\pi/(2 \cdot d_{min})$.

- $\pi/2 < S_i \leq \pi$. In this case, $\rho > d_i$, and an upper bound is readily found for the addend in (2.16) corresponding to reference view i:

$$\frac{S_i}{\rho} \leq \frac{\pi}{d_{min}}. \tag{2.19}$$

Thus, the contribution of each of the reference views to the sum is upper-bounded by the higher of the two bounds in (2.18) and (2.19), which is $\frac{\pi}{d_{min}}$. The mean of all the individual contributions is therefore bounded by this value, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{S_i}{\rho} \leq \frac{\pi}{d_{min}}, \tag{2.20}$$

and inequality (2.16) becomes:

$$k_\omega > \frac{k_v \cdot \pi}{d_{min}}, \tag{2.21}$$

as stated. □

The result in Theorem 2.1 means that we can ensure the stability of the system if we have an estimate of the value of d_{min} for the particular set of reference views we are working with. This estimate does not need to be precise, since what we have found is a fairly conservative bound for the value of k_ω. In practice, the system will be stable with angular velocity gains lower than this threshold. We address next a local stability analysis of our control system.

Proposition 2.2 *The system under the proposed control law (2.10), (2.11) is locally exponentially stable.*

Proof We analyze the behavior of the system locally, i.e., assuming the orientation of the robot has already been corrected ($\alpha_{CG} = \alpha_{CG}^d$). The dynamics of the distance from the goal for the unicycle vehicle considered is then given by:

$$\dot{\rho} = -v\cos\alpha_{CG} = -k_v\frac{1}{n}\sum_{i=1}^{n} S_i. \tag{2.22}$$

Now, taking into account that $S_i \geq \sin S_i$ in all the interval of possible values ($0 \leq S_i \leq \pi$), we have:

$$\dot{\rho} \leq \frac{-k_v}{n} \sum_{i=1}^{n} \sin S_i = -\left[\frac{k_v}{n} \sum_{i=1}^{n} \frac{\sin(\widehat{CR_iG})}{d_i} \right] \cdot \rho. \tag{2.23}$$

It can be readily seen, looking at Fig. 2.6, that for any given current position C of the robot, $\sin(\widehat{CR_iG})$ will be greater than zero for at least one R_i as long as there are at least three reference views (including the goal) and their locations are not collinear. Thus, there exists a positive value λ_{min} such that:

$$\frac{k_v}{n} \sum_{i=1}^{n} \frac{\sin(\widehat{CR_iG})}{d_i} \geq \lambda_{min} > 0. \tag{2.24}$$

From (2.23) and (2.24), it can be concluded that the local convergence to the target state is bounded by an exponential decay, i.e., the system is locally exponentially stable. □

Observe that the actual value of the exponential decay parameter λ_{min} in a particular case will depend on both the geometric distribution of the reference views and the trajectory along which the robot approaches the goal. The less collinear the reference views are, the faster the system will be guaranteed to converge.

2.4 Experimental Results

The performance of the proposed method has been tested both in simulation and with real images.

2.4.1 Simulations

We used the MATLAB® software to obtain the reported results. For the first simulation we present, the reference views were positioned forming a square grid. A randomly distributed cloud of 200 points in 3D was generated and projected in each camera. Four sample homing trajectories with a 25-view reference set and the evolutions of their corresponding motion commands are displayed in Fig. 2.7. The cloud of points is also shown. One of the trajectories starts from a position outside of the grid of reference views. As can be seen, it behaves in the same way as the other three. In all four trajectories, the motion is smooth and the robot converges to the goal position.

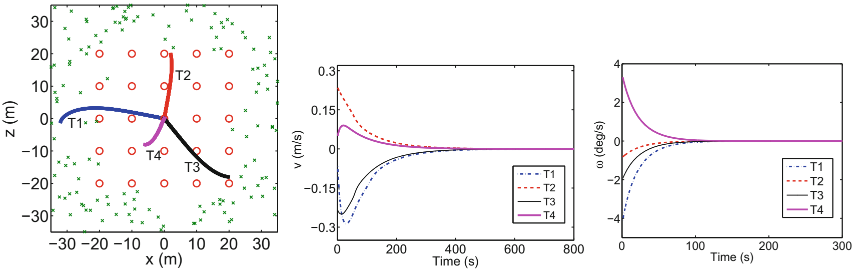

Fig. 2.7 Simulation results. *Left* robot paths for four different trajectories with a reference set consisting of 25 images acquired in positions forming a grid. The goal view is at position (0, 0) for all the trajectories. The locations of the reference images are marked as *circles*. The cloud of 200 points used to compute the 1D trifocal tensors is also shown. *Center* Linear velocity for the four trajectories. *Right* angular velocity for the four trajectories

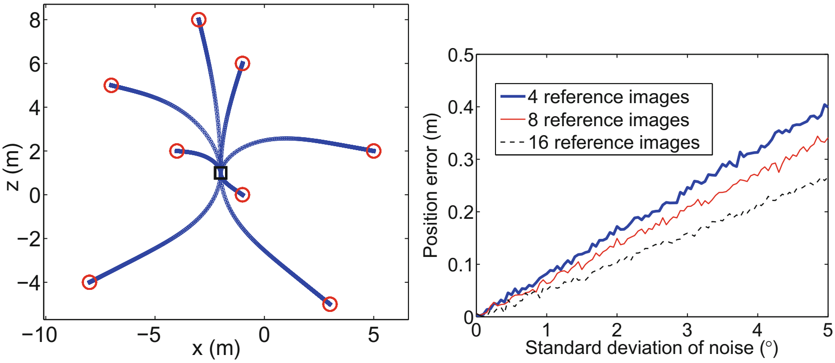

Fig. 2.8 Simulation results. *Left* robot paths with 8 reference images in arbitrary locations. The different paths obtained by taking each of the reference images (marked as *circles*) as the goal, from a common starting location (marked as a *square*), are shown. *Right* final position error as a function of added Gaussian noise for reference sets of 4, 8, and 16 images

The reference views can be laid out in any arbitrary configuration (as long as sufficient geometric diversity on the plane is guaranteed). We illustrate this fact with the simulation shown in Fig. 2.8, left. Eight reference images are used, and they are placed at arbitrary locations in the environment. The plot shows the paths from a common initial location (marked with a square) to the positions associated with the reference images; i.e., a different image is selected as the target each time. We also added Gaussian noise to the angles of the projected points to evaluate the performance of the homing method. Figure 2.8, right, displays the final position error obtained after adding variable noise in simulations with sets of 4 (the minimum number for our method), 8, and 16 reference images. Increasing the number of reference views makes the system more robust to noise, since the linear velocity of the control is computed by averaging out the contributions of all the views.

2.4.2 Experiments with Real Images

In order to assess the performance of our homing method, we tested it with three diverse sets of real omnidirectional images.

2.4.2.1 First Indoor Experiment

The images for this first experiment were obtained in a laboratory setting. The experimental setup is illustrated in Fig. 2.9. It consisted of a nonholonomic unicycle robot base with a catadioptric vision system. The robot used was a Pioneer 3-AT, a four-wheeled platform quite popular in the context of robotics research, suitable for both indoor and outdoor utilization. In addition, the vision system was made up of a Point Grey FL2-08S2C camera and a Neovision HS3 hyperbolic mirror, mounted on top. This catadioptric setup allowed to conveniently obtain omnidirectional images. The imaging system was used without specific calibration other than the assumption that the camera and mirror axis were vertically aligned. We obtained the angular image information defining the optical center in the images and measuring directly the pixel coordinates (see Fig. 2.2). The resolution of the employed images was 800 × 600 pixels.

To generate the reference set of views, 20 images were acquired from locations forming a 5 × 4 rectangular grid with a spacing of 1.2 m, thus covering a total area of 4.8 × 3.6 m^2. In all the tests performed on real images, we used MATLAB® for our computations. We employed an implementation [40] of the SIFT feature extractor, and a RANSAC robust estimation to calculate the 1D trifocal tensors between the views from the matched features. The number of three-view correspondences employed to obtain the trifocal tensor estimations lied in the range of 30–80.

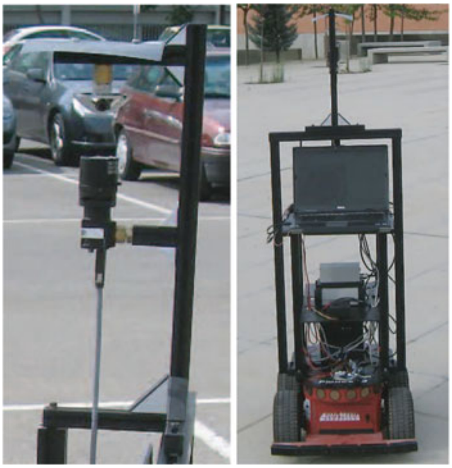

Fig. 2.9 Omnidirectional camera (*left*) and complete setup (*right*) used for the first indoor experiment and the outdoor experiment

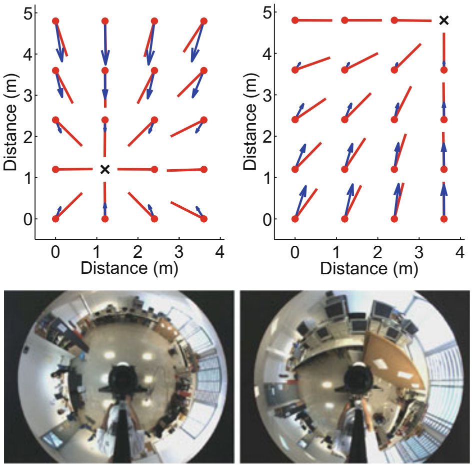

Fig. 2.10 *Top* displacement vectors (*arrows*) and directions of the epipoles (*line* segments) with respect to the goal estimated at every reference position for two different goal locations (marked with a *cross*) in the real setting of the first indoor experiment. *Bottom* goal images corresponding to the homing vector representations on *top*

Although images taken on opposite sides of the room could not be matched, the connections between adjacent or close sets of views were sufficient to recover the relative angles of the complete reference set. Vector field representations for two different goal locations within the grid are displayed in Fig. 2.10.

The arrows in this figure represent displacement vectors and have the following interpretation: If we consider a robot with unicycle kinematic constraints initially situated on each of the reference spots, with its initial orientation aligned with the Y axis of the plot, the arrows represent the displacement that the robot would perform according to the proposed control law. All the vectors have been scaled by an equal factor (the scale is different in the two plots, for better visualization). As can be seen, the magnitude of the vectors becomes larger as the distance to the target increases. Notice in both plots that in the positions from which the angle to the goal is $\pm\pi/2$, the robot does not translate initially. In these cases, it first executes a rotation, due to the cosine term introduced in the linear velocity of the control law (2.10), before moving toward the goal. The line segments in Fig. 2.10 show the estimated directions of the epipoles of the goal position (i.e., the homing vectors) in each of the reference locations. The accuracy of the results obtained in this experiment is remarkable. The outliers in the set of putative point correspondences are rejected through the robust computation of the 1D trifocal tensor, and the availability of a fairly large number of good three-view matches makes it possible to compute the angles between views very precisely.

2.4.2.2 Second Indoor Experiment

In order to assess the performance of our method in more involved scenarios, we tested it on the data set of omnidirectional images used in [41]. The images in this set (see examples in Fig. 2.11) were provided unwrapped and had a resolution of

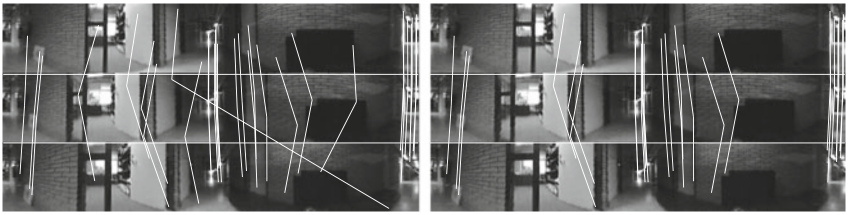

Fig. 2.11 Example of a trio of images from the data set of the second indoor experiment [41] with their putative SIFT correspondences joined by *lines* (*left*). Feature matches remaining after the robust computation of the 1D trifocal tensor (*right*)

720 × 120 pixels. We note here that our approach can be applied indistinctly to both wrapped and unwrapped omnidirectional images. This data set covers an office environment comprising a long corridor (around 14 m of length) and several rooms, and its images were acquired with separations of 0.3 or 0.5 m. The number of images used in this experiment was 213. Despite this denser coverage of the environment with respect to the first experiment, the extraction and matching of SIFT points between the views was much more complicated due to the lower quality of the images. Thus, it was necessary to select the parameters of the SIFT extractor appropriately, which was done following the guidelines described in [9]. A number of 12 three-view correspondences were found to be the minimum necessary for our method to operate on this data set. An example of the matching process for a trio of images from this set, showing the initial SIFT correspondences between them and the matches remaining after the robust computation of the 1D trifocal tensor, is displayed in Fig. 2.11.

The results of this experiment are illustrated in Fig. 2.12. The top three plots in the figure display the homing vectors from all the reference positions to the target location, for three goals chosen in different parts of the environment (inside a room, in the center of the corridor, and near the opposite end of it). As can be seen, the results are fairly accurate and robust despite the low quality of the images. This is also true when the positions are very far away from the goal or even located in a different room. For a small group of locations in the set, it was not possible to compute the angular information and resulting homing vector. These cases occurred when either the number of matched features was insufficient or incoherent results (which were automatically detected and rejected by our method through geometric consistency checks) were obtained.

Obviously, it would be necessary to define intermediate goals in order for the robot to navigate in a setting of this kind. If, for instance, we would like to go from the room on the left (room 1) to the room on the right (room 2) in this environment, the sequence of steps to follow would be: first move through the door out of room 1 to a reachable goal position in the corridor, then travel to the other side of the corridor to a goal position in front of the door of room 2, then get into room 2, and reach the final goal. That is, a global homing task has to be divided into several *direct* homing steps, in each of which the goal location must be directly reachable from the starting position. We illustrate the results of an example of such a task in the bottom plot of

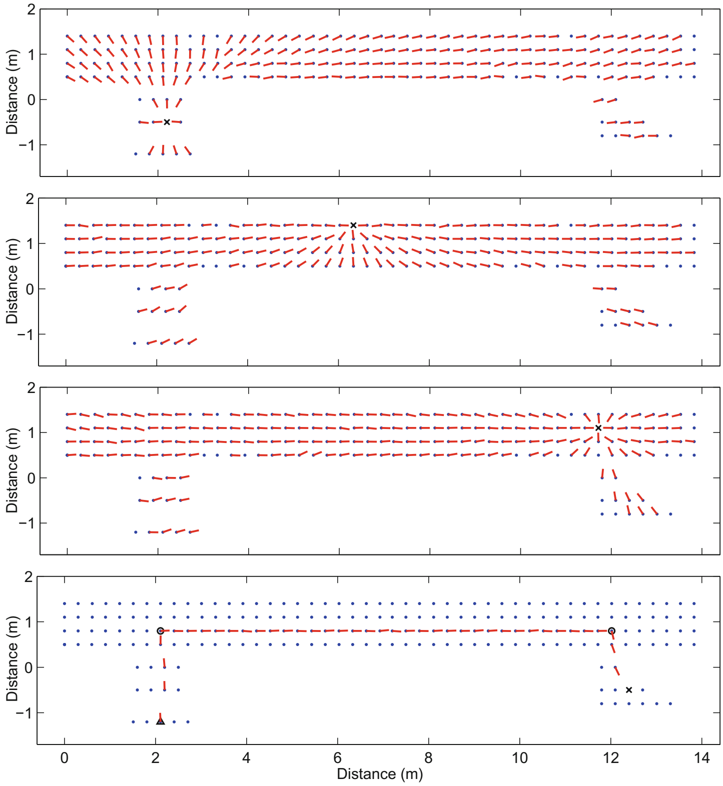

Fig. 2.12 Experimental results for the data set of the second indoor experiment. The environment consists of two rooms and a corridor. *Top three plots* the computed homing vectors from all the positions of the reference views in the set to the goal location, marked with a *cross*, are shown for three different examples. *Bottom plot* sampled path along a chain of reference positions from a starting location (*triangle*) to a homing destination (*cross*) passing through two homing subgoals (*circles*)

Fig. 2.12. The homing vectors along a sampled approximation of the path that the robot would follow are shown. The intermediate goals have been selected manually. Several aspects related with the long-range operation of our homing method are treated in Sect. 2.5.

We also present results with regard to the connectivity of the views in this experiment in Fig. 2.13. Connectivity graphs and their associated adjacency matrices are displayed. As can be seen, only images that are close physically can be initially connected. These links are obtained from the computation of the 1D trifocal tensor between sets of three views. Then, the matrix becomes filled using the indirect angle computation procedure. Two intermediate cases of this process are shown in Fig. 2.13, illustrating how the connections between views are progressively computed. Eventually, almost the complete set becomes interconnected, as shown by the

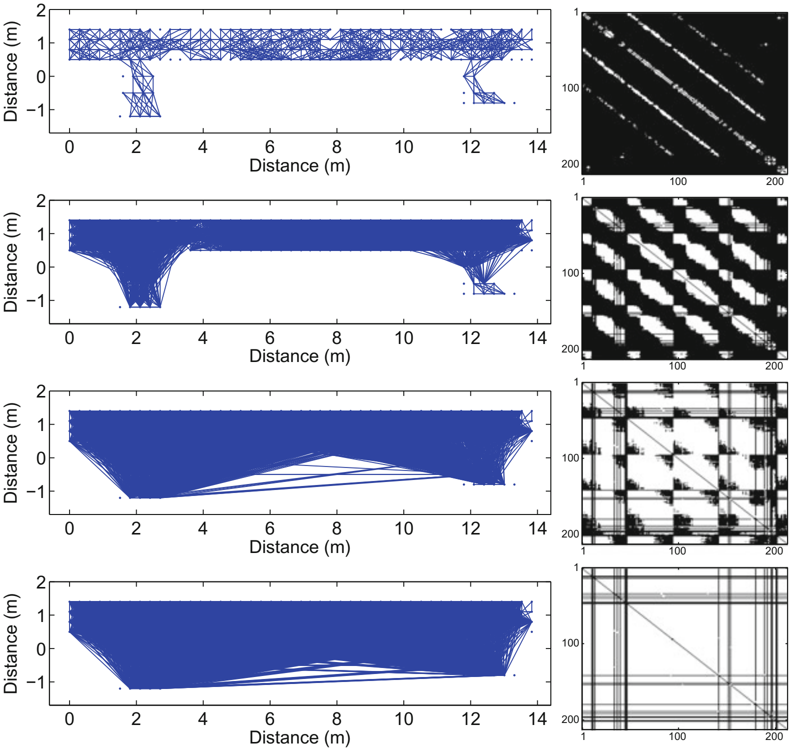

Fig. 2.13 Connectivity between views for the second indoor experiment. *Left column* graph representations of the connectivity, with *lines* joining the connected locations in the environment after using our approach. *Right column* color-coded adjacency matrices corresponding to the connectivity graphs on the *left*. *White color* means “1”, and *black color* means “0”. The values in the axes denote the indices of the images, ranging from 1 to the number of available images. From *top* to *bottom*, four cases are shown: the initial connectivity, two intermediate cases, and the final connectivity results

plots in the bottom row of the figure. The rate of the number of angles computed initially with respect to the total number of angles between all the views is 4.5%: a very low value, due to the setting being large and densely populated with reference images. However, this rate grows gradually to reach a high final value of 82.9%. In the end, only 20 out of the 213 available images could not be connected to the rest of the set, which means that 90.1% of the images were interconnected. These experimental results show the ability of our technique to connect robustly the views between different spaces (e.g., different rooms), through large numbers of intermediate images, and across long distances in a given environment. They also illustrate how our approach can perform well in scenarios with low-quality images and few matched features.

2.4.2.3 Outdoor Experiment

The method was also tested in an outdoor scenario, using a data set acquired in a parking lot at our university campus. The images were obtained using the same robotic platform and camera arrangement described in Sect. 2.4.2.1. The separation between the locations where the different reference images were taken was much larger than in the indoors tests. For this experiment, we used a set of 18 images acquired from positions forming a rectangular grid of approximate size $62 \times 31\ \text{m}^2$. Thus, the distance between opposite corners of the grid was of around 70 m. The minimum distance between two images in this set was 12.5 m. Once again, the SIFT keypoint extractor was employed to obtain feature matches between the images. The number of three-view correspondences used to compute the 1D trifocal tensors was in this case in the range of 20–60. Useful point matches were found mainly on the walls of buildings, in the outer regions of the omnidirectional images. Figure 2.14 shows an example of the three-view matching process for this set, illustrating once again how the robust computation of the 1D trifocal tensor makes it possible to eliminate wrong correspondences.

Similar to what occurred with the indoor tests, in this experiment only the connections (i.e., the angles of the epipoles) between physically close locations could be computed directly, with the remaining angles being calculated indirectly. We computed in this case 40% of the angles between the available views directly. The initial adjacency matrix obtained for this set of images is shown in Fig. 2.15, along with the resulting final adjacency matrix. As can be seen, in this case, we eventually obtained a complete graph (100% of the angles, i.e., we were able to link all the reference positions between one another). The results of the angle calculations for the outdoor experiment are illustrated in Fig. 2.16, which shows the computed homing vectors

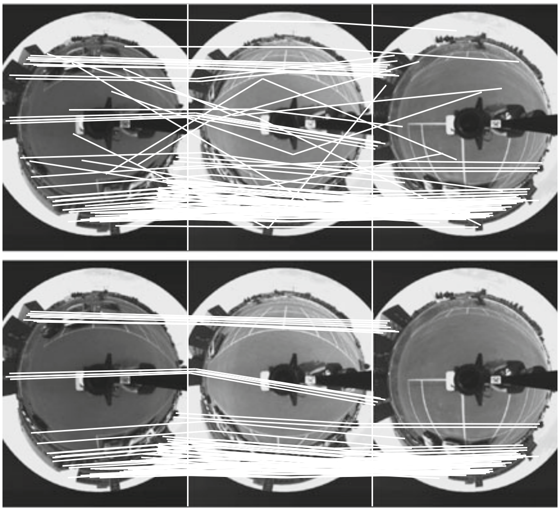

Fig. 2.14 Example of a trio of images from the outdoor data set with their putative SIFT correspondences joined by lines (*top*). Feature matches remaining after the robust computation of the 1D trifocal tensor (*bottom*)

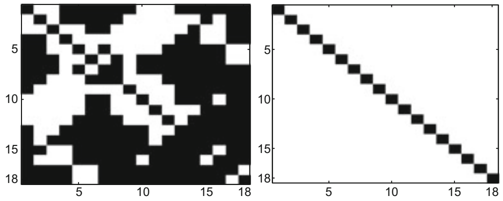

Fig. 2.15 Color-coded adjacency matrices for the outdoor experiment's image set. *White color* means "1", and *black color* means "0". The initial (*left*) and final (*right*) adjacency matrices are shown

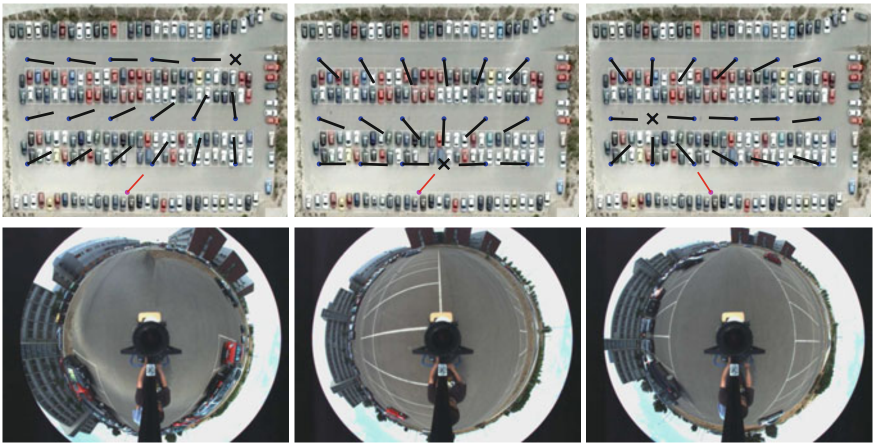

Fig. 2.16 *Top* homing vectors estimated at every reference position for three different goal locations (marked with a *cross*) for the outdoor experiment. The homing vector from an outer position associated with an image not belonging to the reference grid is also shown. The vectors are plotted superimposed on a bird's-eye picture of the real setting. *Bottom* goal images corresponding to the three homing vector representations on *top*

from the reference positions to the goal (marked with a cross) for three different cases. The actual target images corresponding to each case are also shown in the figure. We also show the homing vectors from a position outside of the grid of reference views. This illustrates that as long as it is possible to find sufficient three-view feature matches to compute the 1D trifocal tensor reliably between a given initial image and two of the reference images, our homing method will work, independently of the physical location where that initial image is captured. The setting appears full of cars in the bird's-eye view on which we superimpose our results, but it was much less crowded when the images were acquired. We can see that the results are good in terms of robustness and accuracy. As previously commented in Sect. 2.4.2.2, in order to navigate in this setting, we would need to ensure that the goal location is reachable, e.g., by defining intermediate homing targets such that the robot always moves along the corridors of the parking lot.

The results of the outdoor experiment show that even with a relatively sparse representation of the environment, in terms of the separation between reference images, our approach is still capable of performing robustly. It has potential to cover and interconnect large areas, and allows homing to targets at long distances.

The different experiments we have conducted demonstrate that our method is versatile and has the ability to perform robustly in different scenarios, both indoors and outdoors.

2.5 Discussion

In this section, we compare the characteristics of our homing method with those of other existing approaches and discuss a number of aspects related to it.

2.5.1 Comparison with Existing Work

In the following, we discuss the differences between the approach we present and other closely related works in the fields of visual homing and navigation. It is difficult to make a homogeneous and fair performance comparison among methods, since each of them has very different characteristics. Still, a number of aspects can be addressed. In order to allow a more specific and detailed comparative analysis, let us take at certain points of this discussion the method by Argyros et al. [5] as a representative of visual memory-based approaches, and the work by Goedemé et al. [30] as an example of topological map-based techniques.

The method [5] is, like ours, purely angle-based. The shape of the generated path to the goal depends heavily on the particular distribution of the scene landmarks. In contrast, in our method, the path is independent of this. The difference in behavior is illustrated in the simulation we present in Fig. 2.17. Also, [5] can only be used to replay previously executed homing paths (i.e., it is a pure path-following approach). The method [30] uses an initial homing vector which is refined during the execution of the homing task. This initial vector is computed from the decomposition of the essential matrix for omnidirectional cameras, which is also used for the estimation of the homing direction in the work [31]. We provide in Fig. 2.18 a comparison of the error in the calculation of the homing vector when it is carried out using our method (based on the 1D trifocal tensor) and using the essential matrix. As can be seen, the trifocal tensor based-method is significantly more robust against noise.

As occurs with most existing visual homing approaches, both [5, 30] use only two views to carry out the task, while we use the relations between three views. In these two approaches, the images employed can be acquired in any spatial distribution, whereas in our method, care must be taken to avoid acquiring all the images from positions on a straight line. Both of the related techniques take measures to prevent outlier matches, but these are based on two-view constraints, while we employ a

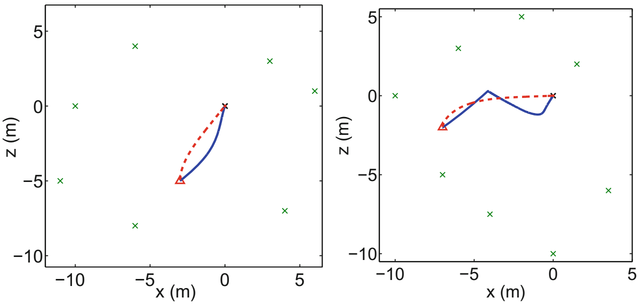

Fig. 2.17 Homing paths from two different initial locations to the goal location situated in (0, 0), with seven (*left*) and eight (*right*) feature matches. The paths generated by the method proposed by Argyros et al. [5] (*solid line*) and the paths generated by the approach presented in this chapter (*dashed line*) are shown

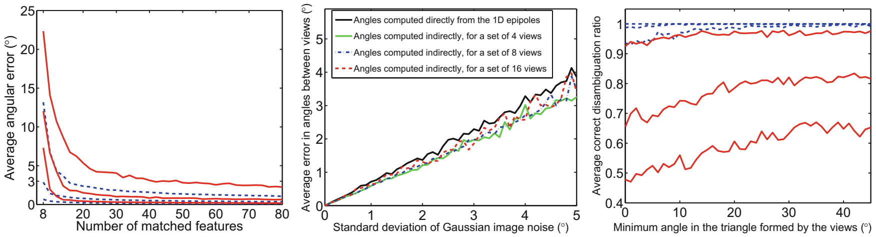

Fig. 2.18 *Left* average error in the computed angle of the homing vector versus number of matched features used. *Solid line*, *top* to *bottom curves* homing vector computed from the essential matrix, with added Gaussian noise of standard deviation 3, 1 and 0.3°, respectively. *Dashed line*, *top* to *bottom curves* homing vector computed through our method based on the 1D trifocal tensor, with added Gaussian noise of standard deviation 3, 1 and 0.3°, respectively. *Center* Error propagation in the computed angles between views, for reference sets of 4, 8, and 16 images. *Right* performance of the procedure to disambiguate the assignment of the epipoles. *Solid line*, *top* to *bottom curves* success ratio of the disambiguation versus degree of alignment of the locations of the views, for seven matched points and Gaussian noise of standard deviation 1, 3, and 5°, respectively. *Dashed line*, *top* to *bottom curves* success ratio for twenty matched points and Gaussian noise of standard deviation 1, 3, and 5°, respectively

stronger, three-view constraint (the 1D trifocal tensor). In contrast with our approach, none of the other two methods takes explicitly into account nonholonomic vehicle kinematics in the design of the motion strategy.

The approach we propose needs a minimum number of four reference images (the goal image and three others). It requires seven three-view correspondences in general configuration (i.e., not all on the same plane of the scene), so that the 1D trifocal tensor can be computed. This minimum number of correspondences is reduced to five if the 1D points are calibrated. Our second indoor experiment shows that our approach is able to operate with a minimum of twelve correct point correspondences between three views, although a number higher than twenty is desirable for better accuracy. In comparison, the method by Argyros et al. [5] employs a minimum of three matches between two views, although in practice they require several tens of correspondences to achieve good performance. The technique by Goedemé et al. [30] needs a minimum of eight points in general configuration matched between two views. Again, more points are usually considered, so as to make the computations more reliable. We believe the information requirements of our method, although higher than those of other existing approaches, are very reasonable, since image feature extractors typically provide many more matches than the minimum needed.

The fact that long-range homing can be carried out directly in our method provides advantages with respect to both image memory-based and topological map-based long-range navigation approaches, since we will require fewer homing subpaths in general, thus increasing the efficiency. We believe that our method can also be more flexible if changes in the path are required: Unlike in [5, 20, 33], the path to a given goal location is not restricted to following a fixed sequence of images, and unlike in [30, 31], all the positions of the reference views may be chosen as direct homing goals at all times.

We believe that two advantages of our method are its robustness, thanks to the use of the trifocal constraint, the geometric consistency checks and the stability properties of the control law, and its accuracy, due to the fact that both the employed visual information and the designed control law are purely angular.

2.5.2 Practical and Performance-Related Considerations

Next, we address several points related to the performance of our method and discuss a series of practical considerations. An interesting issue to tackle is how the error in the computed angles propagates as new angles are obtained using the indirect estimation procedure (Sect. 2.2.2). The theoretical analysis of this aspect in a general case (i.e., for arbitrary angles in the four-view set of Fig. 2.4) turns out to be very involved. We have observed through extensive experiments that the angular error does not propagate significantly across the graph we construct. Moreover, note that large error propagation is prevented by the three-view geometric consistency checks we perform, which ensure that the errors in the angle computations are always maintained within certain bounds. We provide as example the result from a simulation for 4, 8, and 16 images in a square grid-shaped reference set in Fig. 2.18. For this simulation, only the minimum necessary angles were computed directly, and all the rest were obtained indirectly. Notice that the average error does not increase when computing the indirect angles.

Regarding the performance of the angular disambiguation process described in Sect. 2.2.1.2, we have verified through extensive experiments that the procedure we propose has a very high success ratio. Figure 2.18 illustrates some of our results. As can be seen, and for obvious reasons, the disambiguation of the assignment of the epipoles has a larger probability of failure when the positions of the images are closer to being along a straight line. We show in the figure that the epipoles can frequently be assigned wrongly in quite extreme cases (very few points matched, and very high noise). With just a few more points (twenty) available, the success rate improves drastically. We have observed through experimentation that the ambiguity in determining the relative motion between two views from the 1D epipoles (Fig. 2.3) is resolved very effectively by our proposed method. Even for the minimum possible number of points (7) and very high noise (5° of standard deviation in the projected angles), the disambiguation turns out to be correct in 99% of the cases. Note that the incorrect two-view disambiguations that may originate from errors in the assignment of the epipoles are robustly detected and discarded when the joint, three-view coherence of the two-view results is checked (Algorithm 2.1).

The required density of reference images with our method is dictated by the need for a sufficient number of three-view feature matches in order to compute the 1D trifocal tensor reliably. Thus, the maximum separation between the locations of two reference images, or the maximum distance from a given starting location to the locations where the images in the set were taken, is given by the distance at which the number of three-view matches drops below a certain threshold. Also, the positions of the reference images in our method must not all be on a straight line, since both the disambiguation and indirect angle computation procedures rely on the differences between the angles in the triangles created by groups of three views. While acquiring the reference images, it is simple to take this point into account and guarantee that their locations are distributed in the environment with sufficient diversity.

With these conditions in mind, we believe that it would be possible to do the exploration of the environment in order to capture the reference images in a semi-automatic manner. The robot could be run across the environment capturing images either at a certain rate or when the number of matches fell below a certain threshold. A sufficient number of three-view matches between views need to be available, and in practice, the parameters of the feature extractor/matcher have to be tuned in order to adapt to the conditions of the environment and the images. These adjustments, which can be typically done automatically, allow to optimize the number of correspondences. If dynamic changes in the environment occur, our homing approach will be robust to them as long as sufficient features between images can be matched.

The definition of the intermediate homing objectives for our method could be done manually or aided by the use of image classification and place segmentation techniques, in the manner of [42, 43]. In our case, the information that could be used for these purposes is the initial connectivity graph of the environment. For a given view, this graph determines robustly what other views share significant common visual content with it. In order to implement our method, it is required to integrate the homing approach with a reactive obstacle avoidance procedure, as is common in other works in the literature [5, 30].

2.5.3 Summary

In this chapter, we have presented a visual homing method for a robot moving on the ground plane. Employing omnidirectional images acquired from the current robot position, the goal location, and a set of other positions in the environment, the method works by computing the relative angles between all the locations by means of the 1D trifocal tensor. We have proposed a control law to drive the robot to the goal employing the calculated angles. This law has been proven to possess strong stability properties. In addition, the experiments have shown that the approach provides good accuracy and can perform robustly both indoors and outdoors, with low image quality and with varying density of images in the reference set. Homing between distant locations and different settings is also feasible. The online operation of the method is fast, the feature extraction process being the main limitation in terms of speed. The method can be directly applied in settings where stored image databases are available.

A particularly interesting point regarding the methodology proposed in this chapter concerns its application to multirobot behaviors. It is possible to use the described technique in a scenario where a team of agents are required to reach a set of positions in an environment. Each agent can use its own database of images and choose any of them as its target image (or, equivalently, its target position). We believe that this gives rise to attractive possibilities. For instance, the robots may generate, or update, their databases online by exchanging their captured visual information with neighboring robots, using communications. This does not even require that full images are transmitted. It suffices to exchange the list of extracted features, and their descriptors, for each image. This way, communication bandwidth requirements are reduced and unnecessary replication of computation (i.e., image analysis for feature extraction) on multiple agents is avoided. By using images captured online, it is ensured that the visual information used by the robots for the control task is up-to-date. In a scheme of this type, the robots can flexibly decide, in a cooperative fashion, on the assignment between robots and desired positions to fit optimality criteria (e.g., reduce the distances to be traveled). Another interesting point is that the aggregate visual information obtained by the team can provide a dense visual representation of the 2D workspace. This provides flexibility, in the sense that there are many possible candidate target positions spread over the environment, toward which the robots can navigate. All these ideas illustrate that the technique proposed in this chapter can be useful to address relevant multirobot tasks such as coverage of an environment or formation control.

References

1. DeSouza GN, Kak AC (2002) Vision for mobile robot navigation: a survey. IEEE Trans Pattern Anal Mach Intell 24(2):237–267
2. López-Nicolás G, Mezouar Y (2014) Visual control of mobile robots. Robot Auton Syst 62(11):1611–1612
3. Lambrinos D, Möller R, Labhart T, Pfeifer R, Wehner R (2000) A mobile robot employing insect strategies for navigation. Robot Auton Syst 30(1–2):39–64
4. Weber K, Venkatesh S, Srinivasany MV (1998) Insect-inspired robotic homing. Adapt Behav 7(1):65–97
5. Argyros AA, Bekris KE, Orphanoudakis SC, Kavraki LE (2005) Robot homing by exploiting panoramic vision. Auton Robot 19(1):7–25

6. Möller R, Vardy A, Kreft S, Ruwisch S (2007) Visual homing in environments with anisotropic landmark distribution. Auton Robot 23(3):231–245
7. Stürzl W, Mallot HA (2006) Efficient visual homing based on Fourier transformed panoramic images. Robot Auton Syst 54(4):300–313
8. Möller R, Krzykawski M, Gerstmayr L (2010) Three 2D-warping schemes for visual robot navigation. Auton Robot 29(3–4):253–291
9. Churchill D, Vardy A (2008) Homing in scale space. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1307–1312
10. Liu M, Pradalier C, Pomerleau F, Siegwart R (2012) Scale-only visual homing from an omnidirectional camera. In: IEEE international conference on robotics and automation, pp 3944 –3949
11. Yu SE, Kim D (2011) Landmark vectors with quantized distance information for homing navigation. Adapt Behav 19(2):121–141
12. Fu Y, Hsiang TR, Chung SL (2013) Multi-waypoint visual homing in piecewise linear trajectory. Robotica 31(3):479–491
13. Hong J, Tan X, Pinette B, Weiss R, Riseman EM (1992) Image-based homing. IEEE Control Syst Mag 12(1):38–45
14. Lim J, Barnes N (2009) Robust visual homing with landmark angles. In: Robotics: science and systems
15. Basri R, Rivlin E, Shimshoni I (1999) Visual homing: surfing on the epipoles. Int J Comput Vis 33(2):117–137
16. Chesi G, Hashimoto K (2004) A simple technique for improving camera displacement estimation in eye-in-hand visual servoing. IEEE Trans Pattern Anal Mach Intell 26(9):1239–1242
17. López-Nicolás G, Sagüés C, Guerrero JJ, Kragic D, Jensfelt P (2008) Switching visual control based on epipoles for mobile robots. Robot Auton Syst 56(7):592–603
18. López-Nicolás G, Becerra HM, Aranda M, Sagüés C (2011) Visual navigation by means of three view geometry. In: Robot 2011 workshop. Sevilla, Spain
19. Chen J, Dixon WE, Dawson M, McIntyre M (2006) Homography-based visual servo tracking control of a wheeled mobile robot. IEEE Trans Robot 22(2):407–416
20. Courbon J, Mezouar Y, Martinet P (2008) Indoor navigation of a non-holonomic mobile robot using a visual memory. Auton Robot 25(3):253–266
21. López-Nicolás G, Guerrero JJ, Sagüés C (2010) Multiple homographies with omnidirectional vision for robot homing. Robot Auton Syst 58(6):773–783
22. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge
23. López-Nicolás G, Guerrero JJ, Sagüés C (2010) Visual control through the trifocal tensor for nonholonomic robots. Robot Auton Syst 58(2):216–226
24. Shademan A, Jägersand M (2010) Three-view uncalibrated visual servoing. In: IEEE/RSJ international conference on intelligent robots and systems, pp 6234–6239
25. Faugeras O, Quan L, Sturm P (2000) Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera. IEEE Trans Pattern Anal Mach Intell 22(10):1179–1185
26. Quan L (2001) Two-way ambiguity in 2D projective reconstruction from three uncalibrated 1D images. IEEE Trans Pattern Anal Mach Intell 23(2):212–216
27. Dellaert F, Stroupe AW (2002) Linear 2D localization and mapping for single and multiple robot scenarios. In: IEEE international conference on robotics and automation, pp 688–694
28. Guerrero JJ, Murillo AC, Sagüés C (2008) Localization and matching using the planar trifocal tensor with bearing-only data. IEEE Trans Robot 24(2):494–501
29. Becerra HM, López-Nicolás G, Sagüés C (2010) Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. Robot Auton Syst 58(6):796–808
30. Goedemé T, Nuttin M, Tuytelaars T, Van Gool L (2007) Omnidirectional vision based topological navigation. Int J Comput Vis 74(3):219–236
31. Booij O, Terwijn B, Zivkovic Z, Kröse B (2007) Navigation using an appearance based topological map. In: IEEE international conference on robotics and automation, pp 3927–3932

32. Franz MO, Schölkopf B, Georg P, Mallot HA, Bülthoff HH (1998) Learning view graphs for robot navigation. Auton Robot 5(1):111–125
33. Cherubini A, Chaumette F (2011) Visual navigation with obstacle avoidance. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1593–1598
34. Åström K, Oskarsson M (2000) Solutions and ambiguities of the structure and motion problem for 1D retinal vision. J Math Imaging Vis 12(2):121–135
35. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
36. Shashua A, Werman M (1995) Trilinearity of three perspective views and its associated tensor. In: International conference on computer vision, pp 920–925
37. Oriolo G (2014) Wheeled robots. In: Encyclopedia of systems and control. Springer, London, pp 1–9
38. Khalil HK (2001) Nonlinear systems, 3rd edn. Prentice Hall, Englewood Cliffs
39. Slotine JJE, Li W (1991) applied nonlinear control. Prentice Hall, Englewood Cliffs
40. Vedaldi A, Fulkerson B (2008) VLFeat: an open and portable library of computer vision algorithms. http://www.vlfeat.org/
41. Booij O, Zivkovic Z, Kröse B (2006) Sparse appearance based modeling for robot localization. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1510–1515
42. Zivkovic Z, Booij O, Kröse B (2007) From images to rooms. Robot Auton Syst 55(5):411–418
43. Zivkovic Z, Bakker B, Kröse B (2006) Hierarchical map building and planning based on graph partitioning. In: IEEE international conference on robotics and automation, pp 803–809

Springer

http://www.springer.com/978-3-319-57827-9

Control of Multiple Robots Using Vision Sensors
Aranda, M.; Lopez-Nicolas, G.; Sagüés, C.
2017, XIII, 187 p. 75 illus., 58 illus. in color., Hardcover
ISBN: 978-3-319-57827-9