Autonomous Robots 19, 7–25, 2005

Robot Homing by Exploiting Panoramic Vision

ANTONIS A. ARGYROS
Institute of Computer Science (ICS), Foundation for Research and Technology - Hellas (FORTH), Heraklion, Crete, Greece
argyros@ics.forth.gr

KOSTAS E. BEKRIS
Department of Computer Science, Rice University, Houston, Texas
bekris@cs.rice.edu

STELIOS C. ORPHANOUDAKIS
Institute of Computer Science (ICS), Foundation for Research and Technology - Hellas (FORTH), Heraklion, Crete, Greece
orphanou@ics.forth.gr

LYDIA E. KAVRAKI
Department of Computer Science, Rice University, Houston, Texas
kavraki@cs.rice.edu

Abstract. We propose a novel, vision-based method for robot homing, the problem of computing a route so that a robot can return to its initial "home" position after the execution of an arbitrary "prior" path. The method assumes that the robot tracks visual features in panoramic views of the environment that it acquires as it moves. By exploiting only angular information regarding the tracked features, a local control strategy moves the robot between two positions, provided that there are at least three features that can be matched in the panoramas acquired at these positions. The strategy is successful when certain geometric constraints on the configuration of the two positions relative to the features are fulfilled. In order to achieve long-range homing, the features' trajectories are organized in a visual memory during the execution of the "prior" path. When homing is initiated, the robot selects Milestone Positions (MPs) on the "prior" path by exploiting information in its visual memory. The MP selection process aims at picking positions that guarantee the success of the local control strategy between two consecutive MPs. The sequential visit of successive MPs successfully guides the robot even if the visual context in the "home" position is radically different from the visual context at the position where homing was initiated. Experimental results from a prototype implementation of the method demonstrate that homing can be achieved with high accuracy, independent of the distance traveled by the robot. The contribution of this work is that it shows how a complex navigational task such as homing can be accomplished efficiently, robustly and in real-time by exploiting primitive visual cues. Such cues carry implicit information regarding the 3D structure of the environment. Thus, the computation of explicit range information and the existence of a geometric map are not required.

Keywords: robot homing, omni-directional vision, panoramic cameras, vision-based robot navigation

1. Introduction

The goal of this research is to propose a minimalist, yet robust, vision-based method that supports robot homing and to describe a real time implementation of this method on a robotic platform. Homing is a term that robotics has borrowed from biology (Franceschini et al., 1992; Weber et al., 1998), where it is usually used to describe the ability of various living organisms to return to their nest after having distanced themselves to forage. Homing is also a useful behavior for mobile robots. It is assumed that a mobile robot starts moving at a position in its workspace, which we will call the "home" position and executes an arbitrary path, called the "prior" path. The execution of the prior path may have been the result of a different behavior such as workspace exploration or target following. Then, the problem is to develop a means of enabling the robot to find its way back to the home position. Homing accuracy can be crucial, particularly when a subsequent task depends on the accurate positioning of the robot. For example, docking for battery recharging may require homing with high (i.e. in the order of a centimeter) positional accuracy. One of the objectives of this work is to reach this level of homing accuracy in order to provide support for such tasks.

1.1. Related Work on Robot Homing

Robot homing is an instance of the general robot navigation problem. There have been many efforts towards solving navigational tasks using non-visual sensors. Mobile robots are typically equipped with odometry sensors which, in principle, suffice to support homing. However, the errors in the information provided by odometry sensors are considerable and, most importantly, cumulative. Recent research efforts use odometry only as a coarse indication of the robot's pose and employ range (ultrasound or laser) sensors to create or use a metric representation of the environment (e.g. occupancy-grid maps) (Thrun, 1999, 2000; Thrun et al., 2000). Several of these methods have been proven successful in the context of demanding applications in complex environments (Burgard et al., 2002; Trahanias et al., to appear). However, indoor environments often exhibit similar 3D structure at completely different locations, which results in high uncertainty in the estimation of the robots' pose. Therefore, a lot of research effort has been devoted to dealing with uncertainty in robot localization (Burgard et al., 1997; Fox et al., 1998, 1999; Gutmann et al., 1998; Thrun et al., 2000).

On the other hand, visual sensors provide dense information regarding "where is what". A survey of methods for vision-based robot navigation is presented in DeSouza and Kak (2002). When visual sensors are used, topological workspace representations are usually constructed (Choset and Burdick, 2000; Santos-Victor et al., 1999). An interesting approach for homing (Basri et al., 1998) is based on recovering the epipolar geometry relating two images. The applicability of this approach is limited by the difficulty in establishing feature correspondences between distant viewpoints. In order to solve the correspondence problem between distant views, several efforts exploit algorithms that are invariant to affine transformations, such as the Fourier-Mellin transform (Rizzi et al., 2000). Methods that combine range and visual information have also been proposed (Facchinetti and Hügli, 1994; Baltzakis et al., 2003).

The aforementioned methods have been applied to visual input provided by conventional cameras with limited field of view. Panoramic cameras, however, can provide up to a 360° field of view (see Fig. 1). For navigational tasks a wide field of view is essential since a robot can observe most of its surroundings without the need for elaborate gaze control. Furthermore, the likelihood of detecting reference features and the "life-cycle" of a tracked feature are increased, while changes due to dynamic aspects of the environment can be tolerated (Cassinis et al., 1996). Chahl et al. (1997) have used panoramic cameras for egomotion estimation, range computation and localization.

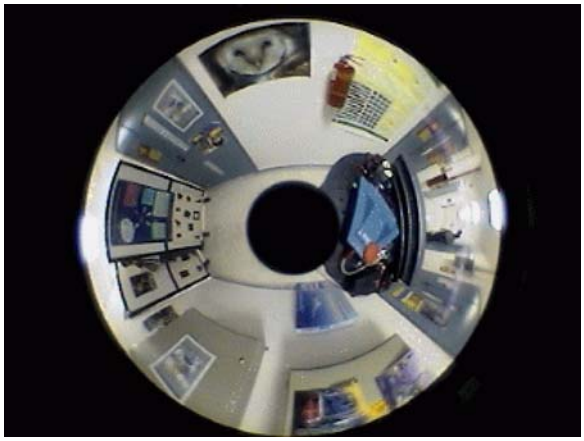


Figure 1. An example of a panoramic image.

Kröse et al. (2002) have studied the problem of robot localization while Bianco and Zelinsky (1999) have investigated possible landmark definitions that allow for robot navigation. Winters et al. (1999, 2000) have used omni-directional cameras to achieve robot navigation by constructing a topological representation of the environment and by applying visual path following.

The choice of vision as the primary source of sensory information for achieving a homing behavior is also inspired by nature, which provides a plethora of visual systems that are successful in assisting long-range navigation and homing. Insects such as ants and bees exhibit remarkable homing abilities based on a minimalist cognitive architecture that associates motion vectors to visual content, without 3D reconstruction of the environment (Dyer, 1996). One of the first algorithmic models for homing was developed to explain the navigational capabilities of ants and bees. Collett (1996), Collett and Rees (1997), and Srinivasan et al. (1996) have studied insect navigation and they proposed various ways in which insects use familiar landmarks on their trip to the home position. Cartwright and Collett (1983, 1987) have proposed the so-called *snapshot model*. According to this model, a snapshot is taken at the home position. When the agent is displaced to a different position, both the perceived angles and the landmarks sizes on the retina change. As a result, the snapshots acquired at these two positions vary significantly in appearance. Pairing sectors of the current and home snapshots using compass information can derive a home vector, pointing approximately to the direction of the home position. Each pairing generates two unit vectors, (a) a tangential vector pointing from the snapshot sector towards the current view sector and (b) a radial vector, which points centrifugally, if the apparent size of the current view sector is smaller than the size of its counterpart in the snapshot and vice versa. Summing all unit vectors derives the homing vector. This approach requires the use of compass information and is successful in driving the robot to the goal only if the starting position is in the vicinity of the goal.

Many other research efforts towards solving the problem of vision-based homing in robotics have been inspired by the snapshot model. For example, Lambrinos et al. (2000) introduced the proportional vector model and implemented it on the Sahabot II. The experiments were conducted on a flat plane in the Sahara desert with four black cylinders as highly distinctive landmarks. The snapshots were aligned at a direction obtained from the polarized light compass of Sahabot. Möller (2000) describes the average landmark vector technique and its implementation on analog hardware, which is closely related to the snapshot model. Franz et al. (1997, 1998a, 1998b) proposed the *average displacement vector* model that is based on the assumption that landmarks are equidistant from the current viewpoint. According to this model, the equal distance assumption is not supposed to drastically affect the convergence properties of their control algorithm. Franz and Mallot (1998) provide an excellent review of biomimetic robot navigation. Most of these approaches use compass information that facilitates the task of corresponding landmarks in panoramas acquired from different viewpoints.

Another way of achieving homing is by "memorizing" the environment. This is the approach taken by appearance-based methods. Cameras are nicely suited for such approaches, since images are usually stored to represent a location and control is associated with each stored image. The images taken during homing are matched with images acquired and stored during the execution of the prior path. Gaussier et al. (2000) developed a neural network approach to map perception to action. The robot computes an array of maximum intensity averages along the horizontal axis. The set of maximum values in the array defines a "place", which a neural network associates with a control that eventually moves the robot to its final destination. Matsumoto et al. (2000) introduced the VSRR ("view-sequence") concept, where the robot stores a sequence of images and associated controls. Then the system computes the displacement in pixels between the current image and the best-matched stored image. This displacement is used in a table to extract controls.

1.2. The Proposed Approach to Homing

We assume a robot equipped with a panoramic camera mounted on it. The basic building block of the proposed homing strategy is a local, reactive control strategy, which enables a robot to move between two adjacent positions. The success of the control strategy depends on the existence of at least three matched image features between the two panoramas acquired at these two positions. The features employed in our approach are image corners, which can be automatically extracted and localized in images (Shi and Tomasi, 1993). Usually, a plethora of such features can be extracted in images acquired in realistic environments as opposed

to other more complex landmarks (Thompson et al., 1999). Establishing feature correspondences in images acquired from adjacent viewpoints is an extensively studied problem in computer vision. This problem can be simplified by tracking the image features in all intermediate frames that the robot acquires between two nearby positions during the execution of the prior path. Therefore, short-range homing, i.e. homing initiated at a position close to home, can be achieved by direct application of the local control strategy. In the case of long-range homing prominent features are greatly displaced and/or occluded, and the correspondence problem becomes much more difficult to solve. In many interesting cases the visual context at the home and the current positions are totally different (e.g. two positions in different rooms) which makes the establishment of correspondences an impossible task, simply because such correspondences do not exist. Therefore, the local control strategy cannot support long-range homing. To overcome this problem, the proposed method decomposes homing into a series of simpler navigational tasks, each of which can be implemented using the proposed local control strategy.

The operation of the overall homing strategy is depicted in Fig. 2. As the robot departs from its home position H it acquires panoramic views of the environment from which it extracts characteristic features and tracks them in subsequent frames. In our implementation we are employing the KLT algorithm for corner tracking (Shi and Tomasi, 1993; Tomasi and Kanade, 1991). As the robot moves, some of the features disappear while new ones appear due to the changing viewpoint, possible changes in lighting conditions, occlusions, etc. The system builds a visual memory that records the

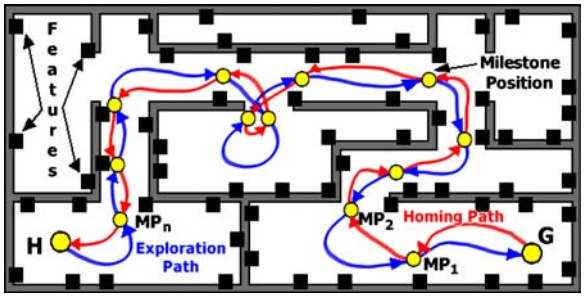


Figure 2. An illustration of long range-homing. The robot starts at position H and executes a "prior" path that leads it to some position G. Then, homing is initiated at G, aiming at guiding the robot back to its home position H.

"life-cycle" of all detected and tracked features. If at a certain moment in time the robot is at location G and decides to return back to its home position H, it automatically defines a number of Milestone Positions (MPs) on the original path connecting H and G, as well as the set of features that will be used to drive the robot between consecutive MPs. The basic aim of this selection process is to guarantee that (a) the robot can move between successive MPs based on the properties of the local control strategy and (b) that visiting all successive MPs will lead the robot to its home position H.

The work presented in this paper exploits panoramic vision and is also influenced by the studies on insect navigation by Cartwright and Collett (1983, 1987). Compared to the existing approaches to robot homing, the proposed method has a number of attractive properties. The proposed method enables a robot to solve the homing problem regardless of the length of the prior path and with minimal computational and sensory requirements. In particular, the main contribution of this work is that it shows that robust, long-range homing capabilities can be achieved by a vision-based approach which uses a simple control strategy and does not make use of a geometric map, of range or of compass information. The local control strategy does not require the definition of two types of motion vectors (tangential and centrifugal), as in the original snapshot model and, therefore, the definition of motion vectors is simplified. Furthermore, there is a simple and intuitive stopping criterion that marks the reaching of the home position. The reactive nature of the employed local control strategy enables a robot to perform other important navigational tasks while homing. For example, the robot motion vector suggested by the local control strategy can always be combined with the motion vector suggested by a reactive obstacle avoidance module. Last but not least, we provide a complete architecture and an implementation for long-range homing. We show how short-range homing strategies can be incorporated in a global framework even in the case that no feature is commonly visible at the home position and the position where homing was initiated.

The proposed method for robot homing has been implemented and extensively tested on a robotic platform in a real indoor office environment. The home position is achieved with high accuracy after long journeys during which the robot performs complex maneuvers. No modification of the environment is necessary to facilitate the robot in its homing task. The proposed method can efficiently achieve homing as long

as enough corners exist in the environment. Due to the simplicity of the tracked features, however, it is guaranteed that there are many of them in a typical indoor environment.

1.3. Overview

The rest of this paper is organized as follows. In Section 2 we present the local control law that is able to move the robot between two adjacent positions. We also provide an analysis of the control strategy demonstrating that the set of positions that are reachable from a specific starting point depends on the spatial configuration of the employed features. In Section 3 the construction and the employment of the visual memory for long-range homing are described. Section 4 presents important issues regarding the detection and tracking of image features in panoramic images while Section 5 summarizes the results from several experiments. Finally, in Section 6, various aspects of the proposed method are discussed and basic conclusions of this research are drawn.

2. Local Control Strategy

We now present the proposed local control strategy that drives a robot between two adjacent positions S and T,[1] provided that there is an established correspondence between at least three features in the panoramic views acquired at these positions. These positions will later be used as milestone positions (MPs) for long-range homing.

2.1. Method Description

Let F_i represent a feature in the environment. We define $A_P(F_i)$ to be the bearing angle of feature F_i as observed from position P, i.e. the direction at which the robot perceives the feature relative to its orientation. A pair of features F_1 and F_2, as observed from position S, define an angle $F_S(F_1, F_2) = A_S(F_2) - A_S(F_1)$ measured in the range $[0..2\pi)$. The locus of positions that can observe the features F_1 and F_2 at angle $F_S(F_1, F_2)$ is the circular arc with end-points F_1 and F_2 that passes through S (see the corresponding solid circular arc in Fig. 3).

A robot motion vector M_{12} is defined so that it lies on the bisector of the angle $\angle F_1 S F_2$ and its magnitude is proportional to the angle difference $|F_T(F_1, F_2) - F_S(F_1, F_2)|$. The direction of M_{12} can easily be computed as $A_S(F_1) + F_S(F_1, F_2)/2$. Intuitively, the partial motion vector M_{12} contributes towards equalizing the corresponding angles at positions S and T. Therefore, the larger the angular difference is, the strongest the component of motion towards this direction will be. The direction of motion is determined by the relative size of the angles; if the angle at T is larger (smaller) than the corresponding angle in S, then the robot moves on the bisector of the angle, towards (opposite to) T. The path that the robot will follow is the branch of the hyperbola that passes through S and has the features F_1 and F_2 as its foci. The robot will stop moving when it reaches a position which belongs to the circular arc $F_1 T F_2$, since for all points T′ on this arc: $|F_{T'}(F_1, F_2) - F_T(F_1, F_2)| = 0$. Each of the branches of the hyperbola with F_1 and F_2 as foci, intersect all circles passing through F_1 and F_2; consequently it is guaranteed that the robot will both arrive at a point with this property and will stop there.

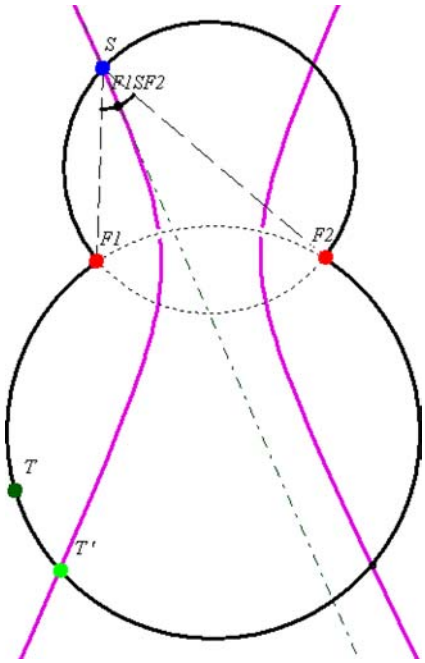


Figure 3. If at position S the robot follows the bisector of the angle formed by two features F_1 and F_2 in the environment, it will effectively move on the branch of the hyperbola that passes through S and has points F_1 and F_2 as its foci.

It is clear that, given only the bearing angles of two features F_1 and F_2, the robot cannot move from a given point S to a target position T. However, if another feature F_3 is considered, then T is constrained to lie

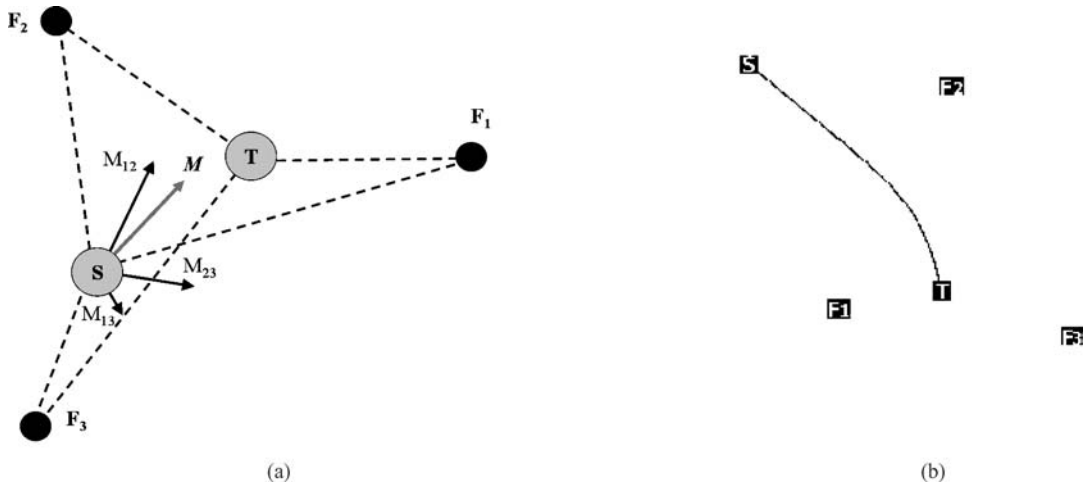


Figure 4. Graphical sketch of the control strategy. (a) The three vectors M_{12}, M_{23} and M_{31} are partial contributions to the global motion vector $\boldsymbol{M}$ defined at position S. (b) The black trace corresponds to the path traveled by a simulated robot between the start position S and the target position T. Numbered rectangles correspond to the three features employed by the local control strategy.

on two more circles, namely the one defined by features F_1, F_3 and position T and the one defined by features F_2, F_3 and position T. Let us now assume that for each pair of features F_i and F_j ($1 \leq i, j \leq 3$, $i \neq j$), we define motion vectors M_{ij} as before and force the robot to move in the direction of the vector sum $\boldsymbol{M}$ of all partial motion vectors M_{ij}. Figure 4(a) illustrates the quantities involved in the proposed control strategy. Figure 4(b) gives an example of the behavior achieved by a simulated robot utilizing the proposed control strategy. Note that the trace of the robot towards the goal position is not a straight line. This indicates that at each position, the global motion vector $\boldsymbol{M}$ is not necessarily directed towards the target position T. Still, for the case of Fig. 4(b), as the robot adjusts its motion vector $\boldsymbol{M}$ at every new position, it converges to the goal position. Whether the target position has been reached can easily be verified by checking whether the magnitude of the motion vector $\boldsymbol{M}$ is zero. Practically, this is achieved by comparing the magnitude of $\boldsymbol{M}$ against a predefined, positive threshold that represents the desired tolerance in homing accuracy.

2.2. *Properties of the Local Strategy*

Given the definition of the motion vector $\boldsymbol{M}$, two questions need to be answered in order for $\boldsymbol{M}$ to constitute an appropriate control vector for moving the robot from S to T:

(a) *Assuming that the robot's trajectory passes through T, will the robot stop there*?
Assuming perfect sensing, T is the only location on the plane from where three given features can be observed with a specific set of angles. Therefore, if the robot arrives at T, it will stop there because this is the only location on the plane where all M_{ij} and, therefore, $\boldsymbol{M}$ are equal to zero. A degenerate case arises when the target position T and the three features belong to the same circle. This special case is discussed later in this section.

(b) *Given a starting position S, which is the locus of positions that are reachable by following the motion vector* $\boldsymbol{M}$?
Simulations have been carried out to demonstrate the convergence properties of this control strategy. To do so, the definitions of the *reachable* and *catchment areas*[2] are introduced.

Definition 1. Let S be a point on the plane and F_1, F_2 and F_3 be three point features on the plane. The reachable area R (S, F_1, F_2, F_3) is defined as the set of points that the robot can reach starting from S, by employing the bearing angles of features F_1, F_2 and F_3 in the proposed control strategy.

Definition 2. Let T be a point on the plane and F_1, F_2 and F_3 be three features on the plane. The catchment

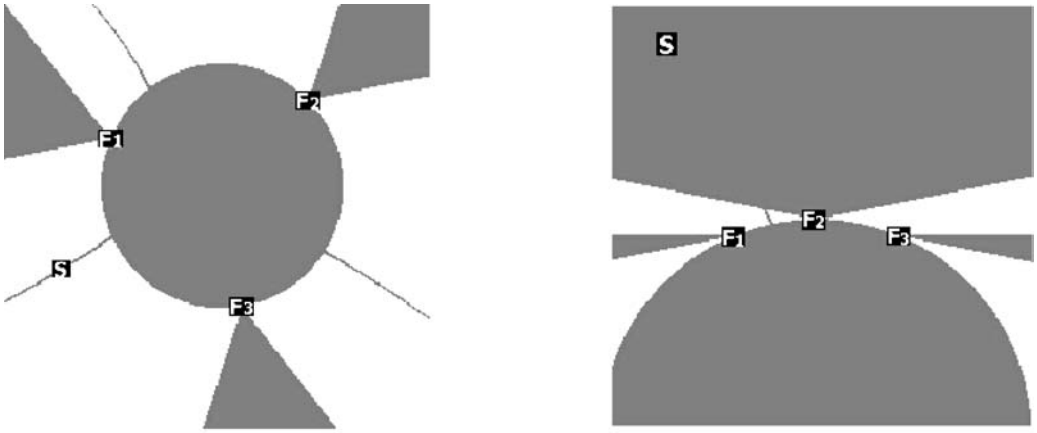


Figure 5. Investigation of reachable areas for the case of three features F_1, F_2 and F_3. Reachable area appears in gray color while unreachable area appears in white color. The starting position is designated by the symbol S, while the observed features are numbered. (a) Reachable area for three features in general configuration, (b) reachable area in the case of almost collinear features.

area C (T, F_1, F_2, F_3) is defined as the set of all starting positions from which T can be reached by employing the bearing angles of features F_1, F_2 and F_3 in the proposed control strategy.

The reachable area contains all reachable destinations of a robot initially placed on S and moving according to the proposed control strategy. The catchment area contains all possible starting points of a robot that has reached position T. Although we do not have an analytical proof on the shape of the reachable area, it has been experimentally validated that the set R (S, F_1, F_2, F_3) consists of:

(a) *Area A1*: the interior of the circle defined by features F_1, F_2 and F_3.
(b) *Area A2*: the intersections of half-planes defined by the sides of the triangle $F_1F_2F_3$.
(c) *Area A3*: curves that emanate from the circle defined by the three features.

Areas A1 and A2 are always reachable, independent of the starting position S. On the other hand, area A3 changes shape depending on S. Figure 5(a) shows illustrative examples. Features F_1, F_2 and F_3 are represented by numbered rectangles. The robot starts moving at the position S of a simulated workspace. The reachable area of this starting position is shown in gray color. White color designates areas that the robot cannot reach.

Several simulations have been conducted to monitor the catchment area of points belonging to areas A1 and A2, in order to obtain further evidence to the fact that the shape of these areas does not depend on the starting position. Towards this end, a goal position is set and the starting robot position is varied. Indeed, the catchment area of every point in areas A1 and A2 is the entire plane, which implies that points in these areas are reachable independently of S.

A special case is encountered when the target position and all features belong to the same circle. In this degenerate case, all points on this circle satisfy the stopping criterion. Another special case is encountered when the employed features are collinear. Then, all points on the plane are reachable apart from the line that connects the set of collinear features. This is because in the case of collinear features, the circle defined by the three features degenerates into a half-plane. The region between the extensions of the sides of the degenerate triangle of features is the other half-plane. Thus, points on either side of the line defined by the collinear features are reachable. Points exactly on this line are not, because then, the target position and the three features are degenerately co-circular. Figure 5(b) illustrates the case of three almost collinear features.

The proposed control strategy can be generalized to employ more than three features. Each feature pair may contribute to the global motion vector $\boldsymbol{M}$. The shape of the reachable area is much more complex when more

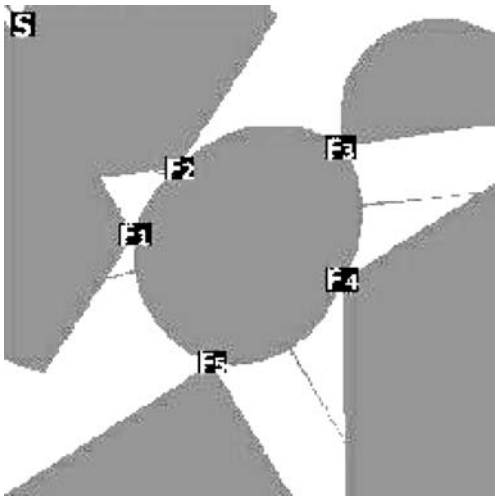


Figure 6. In the case of more than three features, the shape of the reachable area is more complex compared to that of three features. However, points within the convex hull of the features are always reachable.

than three features are considered. However, simulations have shown that if a point lies within the convex hull of the features, then it is guaranteed to be a reachable position. Figure 6 shows the reachable area for point S when a configuration of five features is employed.

An important question is whether, by measuring bearing angles only, the robot can determine if a specific position T is reachable or not. To develop such a criterion, we consider features in a retinotopic order, as observed from position T, i.e. in an order that guarantees that $i > j \Leftrightarrow A_T(F_i) > A_T(F_j)$. Note that the ordering of features depends on the target position T. We then consider the angles $F_T(F_i, F_j)$, $1 \leq i \leq n$, $j = (i + 1) \bmod n$ formed by consecutive features in the defined order. Position T is guaranteed to belong to the convex hull of features $F_1, F_2, \ldots, F_n$ if:

$$\forall i \in \{1, \ldots, n\}, \Phi_T\left(F_i, F_{(i+1) \bmod n}\right) < \pi. \quad (1)$$

Since the convex hull of the features is a subset of the reachable area, the above criterion guarantees that position T is reachable from any starting position. Criterion (1) is sufficient but not necessary, in the sense that there exist reachable positions that do not satisfy it.

2.3. Algorithm for the Local Control Law

We are now in the position to fully describe the algorithm that allows a robot to move from a position S to a position T, given a set of features F_1, F_2, $\ldots$, F_n, $n > 2$, that have been corresponded between views S and T, acquired at positions S and T respectively, and for which the bearing angles and $A_T(F_i)$ are known. Figure 7 presents this algorithm in pseudocode.

3. Long-Range Homing

Assume that the home H and the goal position G do not have any feature in common and, therefore, the local control strategy presented in Section 2 cannot be employed to directly support homing. In order to alleviate this problem, milestone positions (MPs) are introduced. Based on the employed feature selection algorithm, the robot detects features (corners) in the view acquired at its home position. As it departs from this position following the prior path, it continuously tracks these corners in subsequent frames. During its course, some of the initially selected features may be lost while other features may appear. In the first case the system "drops" the features from subsequent tracking. In the second case, features start being tracked. This way, the system builds a "visual memory" where information regarding the "life-cycle" of features is stored. A graphical presentation of this type of memory is shown in Fig. 8. The vertical axis in this figure corresponds to all the features that have been identified and tracked during the execution of the prior path. The horizontal dimension corresponds to time. Horizontal black lines correspond to the life cycle of a certain feature. It is theoretically possible for a feature to disappear at some point in time t, and reappear later in the journey of the robot from home position to position G. In this case, the same environmental feature will appear as two distinct features in the visual memory, but this does not affect the homing procedure.

As soon as homing is initiated at position G, the robot first decides how far the robot can go towards H based on the extracted and tracked features. A position with these characteristics is denoted as MP_1 in Fig. 8. The automatic selection of MP_1 essentially amounts to the problem of finding the three features that are visible at G, have the longest tracked trajectories and their A1 reachability area contains MP_1. In

- **Input:**
  - The angles $A_T(F_i)$ of features F_i ($1 \le i \le n$), observed from the target position T. During algorithm execution, P denotes the current position of the robot.
  - A threshold t for the magnitude of the motion vector $\boldsymbol{M}$ determining whether the target position T has been reached.
- **Output:** Upon termination, the robot will reach position T, if position T is reachable.
- **Algorithm:**
  1. Based on $A_T(F_i)$, decide whether position T is reachable or not, by employing the criterion of eq. (1).
  2. If position T is unreachable, exit**.**
  3. For all $1 \le i \le n$, set j = $(i+1) \bmod n$ and compute $\Phi_T(F_i, F_j)$
  4. Repeat {
     - 4.1 Set motion vector $\boldsymbol{M}$ =(0,0)
     - 4.2 For all $1 \le i \le n$, do:
       - 4.2.1 j = $(i+1) \bmod n$
       - 4.2.2 Compute $\Phi_P(F_i, F_j)$
       - 4.2.3 Define partial motion vectors M_{ij} as described is section 2.1.
       - 4.2.4 Add M_{ij} to $\boldsymbol{M}$.
     - 4.3 Update position P by moving according to motion vector $\boldsymbol{M}$.

     } Until ($\|\boldsymbol{M}\| < t$)

Figure 7. The local control strategy that can move a robot from a start position S to a goal position T, given that at least three features have been corresponded between S and T.

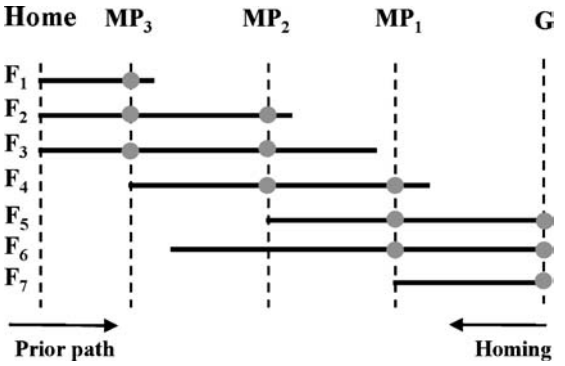


Figure 8. Graphical presentation of the memory built to support homing. The task of homing can be decomposed into a series of simpler navigation tasks, which involve visiting sequentially a number of Milestone Positions (MPs) by employing the local control strategy of Fig. 7.

Fig. 8 these are features F_5, F_6 and F_7, which is a trivial selection since there are no other visible features (we also assume that position MP_1 belongs to the reachability region defined by features F_5, F_6 and F_7). Achieving MP_1 is feasible (by definition) by employing features F_5, F_6 and F_7 in the proposed local control strategy because these features can be tracked (and therefore corresponded) between G and MP_1. The algorithm proceeds in a similar manner to define the next MP towards home. The procedure terminates when the last visited position coincides with the home position.

The local control strategy of Section 2 does not necessarily guarantee the achievement of the orientation with which the robot has previously visited this position. This is because it takes into account the differences of the bearing angles of features and not the bearing angles themselves. This poses a problem in the process of switching from the features that drove the robot to a certain MP to the features that will drive the robot to the next MP. This problem is solved as follows. Assume that the robot has originally visited a position P with a certain orientation and that during homing it arrives at position P', where P' denotes the same position, visited under a different orientation. Suppose that the robot arrived at P' via features $F_1, F_2, \ldots, F_k$.

The bearing angles of these features as observed from position P are $A_P(F_1)$, $A_P(F_2)$, ..., $A_P(F_k)$ and the bearing angles of the same features as observed from P' are $A_{P'}(F_1)$, $A_{P'}(F_2)$, ..., $A_{P'}(F_k)$. Then, it holds that:

$$A_P(F_i) - A_{P'}(F_i) = \varphi, \quad \forall i, 1 \leq i \leq n, \qquad (2)$$

where φ is constant and equal to the difference in the robot orientation at P and P'. This is because panoramic images that have been acquired at the same location but under a different orientation differ by a constant rotational factor φ. Since both $A_P(F_i)$ and $A_{P'}(F_i)$ are known, φ can be calculated. Theoretically, one feature suffices for the computation of φ. Practically, for robustness purposes, all tracked features should contribute to the estimation of φ. Errors can be due to feature tracking inaccuracies and/or due to the non-perfect achievement of P during homing. For the above reasons, φ is computed as

$$\varphi = \text{median}\{A_P(F_i) - A_{P'}(F_i)\}, \quad 1 \leq i \leq n. \qquad (3)$$

Having an estimation of the angular shift φ between the images acquired at P and P', the retinal coordinates of all features detected during the visit of P can be predicted. Feature selection is then applied to small windows centered at the predicted locations. This calculation results in registering all features acquired at P and P' and permits the identification of a new MP and the continuation of the homing procedure.

An important decision is the selection of the number of features that should be corresponded between two consecutive MPs. It has been shown that three features suffice, although more features can be used, if available. It is quite important that only a few (three) matched features are required by the local control strategy; this almost guarantees that there will be a series of MPs that when sequentially visited, will lead the robot to its home position. The advantage of considering more than three corresponded features is that reaching MPs (and consequently reaching the home position) becomes more accurate because feature-tracking errors are smoothed out. However, as the number of features increases, the number of MPs also increases because it is less probable for a large number of features to "survive" for a long period. In a sense, the homing scheme becomes more conservative and it is decomposed into a larger number of safer, shorter and more accurate reactive navigation sessions. Specific implementation choices and related results are discussed in the experiments section of the paper.

4. Panoramic Sensing

In this section we deal with the details pertaining to the detection and tracking of image corners which constitute the perceptual features that are employed by the local control strategy described in Section 2 and used in the construction of the visual memory in Section 3. Image corners have several attractive properties that favor their selection as the primitive visual information employed by the proposed homing method. Specifically, there are many of them in most indoors environments. Moreover, their definition, extraction and tracking are mathematically rigorous and computationally efficient. Nevertheless, the homing strategy can, in principle, exploit any other point feature that can be extracted from panoramic images (e.g. centroids of detected and tracked colored image blobs).

4.1. Extracting and Tracking Features in Panoramic Images

The local control strategy for moving between adjacent positions is based on the availability of feature correspondences between two panoramic views. If the two views have been acquired from significantly different viewpoints, feature correspondence is a non-trivial task (Lourakis et al., 2003). For this reason, the proposed homing strategy achieves the required feature correspondences through feature tracking in a series of panoramic images that the robot acquires as it moves. This guarantees small inter-frame displacement, which, in turn, facilitates the task of feature correspondence. More specifically, we have employed the KLT tracking algorithm (Shi and Tomasi, 1993; Tomasi and Kanade, 1991). KLT starts by identifying characteristic image features, which it then tracks in a series of images.

An important property of KLT is that the definition of features to be tracked guarantees optimal tracking. The definition of features is based on the quantity:

$$Z(i, j) = \begin{bmatrix} \sum\sum g_x^2 & \sum\sum g_x g_y \\ \sum\sum g_x g_y & \sum\sum g_y^2 \end{bmatrix}, \qquad (4)$$

which is defined over an $N \times N$ neighborhood of image point (i, j). In Eq. (4) g_x and g_y are the partial

derivatives of the image intensity function. The eigenvalues λ_1 and λ_2 of the matrix Z are computed. Good features to track are considered those satisfying the rule

$$\min\{\lambda_1, \lambda_2\} > t, \tag{5}$$

where t is a predefined threshold. Tracking is then based on a Newton-Raphson style minimization procedure that minimizes the error vector e:

$$e = \sum\sum [I - J] \begin{bmatrix} g_x \\ g_y \end{bmatrix}, \tag{6}$$

where I and J are the two images containing the features to be tracked. The minimization of e is based on the solution of the linear system

$$Zd = e, \tag{7}$$

where d is the displacement of the tracked feature. In addition to the purely translational model, tracking can take into account the case of an affine image transformation between two consecutive images. Theoretically, the latter is more general and allows tracking of features that have undergone shearing or rotation. Shi and Tomasi (1993) propose the use of the translation model for a good displacement measurement of features and confine the affine model to monitoring a feature's quality by checking the dissimilarity between the initial and the current frame.

The KLT corner detection and tracking is not applied to the panoramic images provided by a panoramic camera (e.g. image of Fig. 1) but rather on a cylindrical version of it (e.g. the image of Fig. 9). Such an unfolding transformation can easily be achieved using a polar-to-Cartesian transformation (Argyros et al., 2004). In the resulting cylindrical image, the full 360° field of view is mapped on the horizontal image dimension. In the remainder of this paper, if not otherwise stated, the term panoramic image refers to a cylindrical one. Once a corner feature F is detected and tracked in a sequence of such images, its bearing angle $A_P(F)$ of the feature can be computed as:

$$A_P(F) = \frac{2\pi x_F}{D}, \tag{8}$$

where x_F is the x-coordinate of feature F in the image acquired at position P, and D is the width of this panoramic image in pixels.

4.2. *Reducing Uncertainty in Feature Tracking*

Feature correspondence may result in feature mismatches which may introduce considerable errors in the computation of the motion vector. One way to eliminate some of these errors is to exploit the ordering of features in panoramas acquired at two different positions. More specifically, features that do not appear in the same order in panoramas acquired at two different positions are excluded from the computation of the global motion vector. In order to detect such features, the Longest Common Subsequence algorithm (Cormen et al., 1996), which is a dynamic programming technique, has been employed. Formally, given two sequences $X = \langle x_1, x_2, \ldots, x_k\rangle$ and $Z = \langle z_1, z_2, \ldots, z_m\rangle$, Z is considered a subsequence of X if there exists a strictly increasing sequence $\langle i_1, i_2, \ldots, i_k\rangle$ of indices of X such that, for all $j = 1, 2, \ldots, k, x_i(j) = z_j$. Given three sequences X, Y and Z, Z is a common subsequence of X and Y, if Z is a subsequence of both X and Y.

Consider now the sequence of features $FS = \langle F_1, F_2, \ldots, F_L\rangle$ extracted at the start position S and the sequence $FT = \langle F'_1, F'_2, \ldots, F'_K\rangle$ extracted at the goal position. The estimation of the motion vector $\boldsymbol{M}$ (from S to T) is based on the features contained in the maximum-length common subsequence of FS and FT. The time complexity of the Longest Common Subsequence Algorithm is $O(L + K)$.

5. Experimental Results

The proposed method has been experimentally tested on LEFKOS, the RWI B21r mobile robotic platform of

Figure 9. The result of unfolding the panoramic image of Fig. 1.

Figure 10. LEFKOS, the mobile robotic platform of ICS/FORTH with a panoramic camera mounted on it.

the Computational Vision and Robotics Laboratory of ICS-FORTH. A panoramic camera has been mounted on top of LEFKOS to provide the necessary panoramic views (see Fig. 10). An 800 MHz Pentium III processor was dedicated to vision processing and robot control. The experimental testing took place in a real office environment. No special arrangements and modifications were made in the environment to facilitate the execution of the proposed homing method. Two series of experiments have been conducted aiming at testing different aspects of the proposed homing strategy. Clearly, the robustness of the overall homing strategy heavily depends on the robustness and the accuracy of the local control strategy that moves the robot between successive MPs. In order to quantitatively assess the robustness and accuracy of the local control strategy a series of related experiments have been conducted. Moreover, a series of long-range homing experiments have been executed.

5.1. Experiments with the Local Control Strategy

The first series of the conducted experiments study the ability of the proposed local control strategy to support robot motion between adjacent positions. In each of these experiments, the robot is placed at a particular home position where it initially detects N features. These features are subsequently tracked as the robot moves away from the home position, executing the prior path. The prior path is executed with the robot performing a purely translational motion with a speed of 15 cm/sec. The execution of the prior path is terminated as soon as the number of tracked features drops below a threshold of M features. Then, the robot employs the proposed local control strategy to return back to its home position. Six different combinations of values for parameters N and M were tried to investigate how the performance of the proposed local control strategy is influenced by feature selection. For each pair of parameters M and N, ten independent navigation sessions were executed resulting in a total of sixty navigation sessions. In all these experiments the home position remained the same; however the directions of the prior paths randomly varied among different experiments. Table 1 summarizes the quantitative results obtained from these experiments.

The first and second rows of Table 1 measure the average length of the prior path and the standard deviation of this length, respectively. It can be verified that as

Table 1. Performance of the proposed local control strategy in supporting robot motion between adjacent positions.

	$N = 50, M = 45$	$N = 50, M = 40$	$N = 50, M = 35$	$N = 100, M = 95$	$N = 100, M = 90$	$N = 100, M = 85$
Mean distance traveled (m)	3.24	4.99	6.40	1.72	3.17	5.71
St. dev. of the distance traveled (m)	0.54	1.27	0.67	0.45	0.83	1.11
Mean homing error (cm)	3.20	8.90	28.40	15.20	16.34	17.43
St. dev. of error (cm)	2.35	10.08	16.97	6.75	7.99	9.23
Average number of features survived until reaching home	39	35	27	84	79	72

we relax the constraint on the number of features that should be successfully tracked before starting homing, the path traveled by the robot increases considerably. This occurs at the cost of decreased positional accuracy for homing (third and fourth row of Table 1). It is interesting to observe that the effect of accuracy degradation is more profound in the case of smaller N. For example, for $N = 50$, varying the number of tracked features from 45 to 35 results in an average of 25 centimeters increase in the error when attempting to reach the home position. On the contrary, when 100 features are employed, varying the number of tracked features from 95 to 85 increases the positional error by approximately 2 centimeters. Analogous observations hold for the standard deviation of the positional error. The fifth row of Table 1 shows the average number of features that are still tracked upon arrival at the home position at the end of the experiment. As it can be verified, the number of features lost during homing is in the same order as the number of features lost during the execution of the prior path. It is important that in all these experiments the robot had to perform a rather complex maneuver including an 180° rotation. This is because the prior path is executed with a purely translational motion, therefore, returning to the home requires 180 degrees of change in robot pose.

The above series of examples show that the proposed control law can achieve a target position with remarkable accuracy which clearly depends on the number of features employed. The results also demonstrate that with corner features, many more that the theoretic minimum of three should be employed. This is not a problem because typically, a lot of corners can be detected in indoor environments. Moreover, it should be stressed that the optimal number of features may vary depending on various parameters such as robot speed, the distribution of features in the environment etc.

The accuracy of the proposed method for robot homing is attributed to the use of panoramic vision. It turns out that the accuracy in reaching a certain position depends on the accuracy in localizing the image features but also on the spatial arrangement of features around the target position. To illustrate this, assume a panoramic view that captures a full 360° view of the environment in a typical 640×480 image. The dimensions of the cylindrical panoramic images produced by such panoramas are 1394×163, which means that each pixel corresponds to 0.258° of the visual field. If a corner detector can localize a corner with an accuracy of 3 pixels, the accuracy of measuring a bearing angle of a feature is 0.775° or 0.0135 radians. This implies that the accuracy in determining the angular extent of a pair of features is 0.027 radians, or, equivalently, that all positions in space that view pair of features within the above bounds cannot be distinguished.

Figure 11 shows results from related simulation experiments. In Fig. 11(a), a simulated robot, equipped with a panoramic camera, observes the features in its environment with the accuracy indicated above. Then the set of all positions that the robot could reach by the proposed control strategy are shown in the figure in dark gray color. It is evident that all such positions are quite close to the true robot location. Figure 11(b) shows a similar experiment but involves a robot that

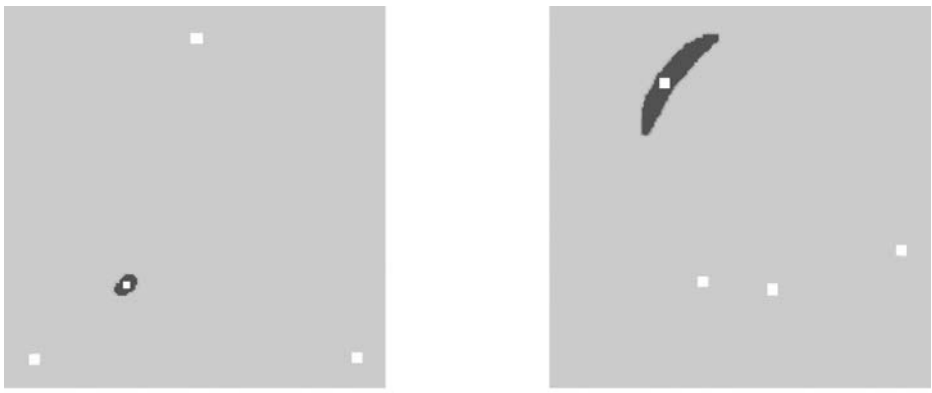


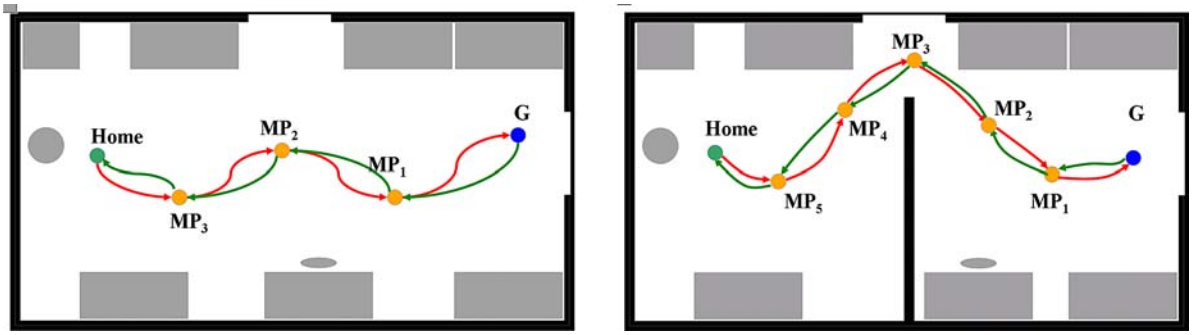
Figure 11. Influence of the arrangement of features on the accuracy of reaching a desired position. Dark gray area represents the uncertainty in position due to the error in feature localization (a) for a panoramic camera and (b) for a 60° f.o.v. conventional camera.

is equipped with a conventional camera with limited field of view that observes three features. Because of the limited field of view, features do not surround the robot. Due to this fact, the fuzziness in determining the true robot location has increased substantially. It is important to note that in the experiment of Fig. 11(b) the camera captures 60° of the visual field in a 640×480 image. Thus, each pixel represents 0.094° of the visual field and the accuracy of measuring a bearing angle of a feature is 0.282° or 0.005 radians. Thus, accuracy in determining the angular extend of a pair of features is 0.01 radians, which is almost three times larger, compared to the accuracy of the panoramic camera. Thus, although feature localization is more accurate in the case of the hypothesized conventional camera, the uncertainty in reaching a desired position is much higher because of the arrangement of features in the visual field of the robot. In the proposed local control strategy, as shown in Section 2.2, each target MP lies within the convex hull of the features that will be used to drive the robot towards it. Since features surround the target position, the accuracy in reaching it is very high. These results indicate that in the context of this work, a sensor that captures a large portion of the visual field is preferable, compared to a sensor that captures more accurately a smaller visual field.

5.2. *Homing Experiments*

Figure 12(a) gives an approximate layout of the robots' workspace and the location of the robot's start position in a representative long-range homing experiment. The robot leaves its home position and, after executing a predetermined set of motion commands, reaches position G, covering a distance of approximately eight meters. Then, homing is initiated, and a number of MPs are automatically defined. The robot sequentially reaches these MPs to eventually reach the home position. Note that the properties of the local control strategy applied to reaching successive MPs are such that the homing path is not identical to the prior path. During this experiment, the robot has been acquiring panoramic views and processing them on-line. Image preprocessing involved unfolding of the original panoramic images and gaussian smoothing ($\sigma = 1.4$). The resulting images were then fed to the KLT corner tracker. Potential features were searched in 7×7 windows over the whole image. For a feature to be considered as a candidate for tracking, threshold t in Eq. (5) was set to 1000. Feature tracking has been accomplished with the aim of image pyramids and the Newton-Raphson iterative minimization scheme. Two pyramid levels have been used. Tracking of a feature was stopped when the determinant of its Z matrix (Eq. (4)) dropped below 0.1. The robot's maximum translational velocity was 4.0 cm/sec and its maximum rotational velocity was 3 deg/sec. These speed limits depend on the image acquisition and processing frame rate and are set to guarantee small interframe feature displacements which in turn, guarantee robust feature tracking performance. The 100 strongest features were tracked at each time. After the execution of the initial path, three MPs were defined so as to guarantee that at least 80 features would be constantly available during homing.

Figure 13 shows snapshots of the homing experiment as the robot reaches the home position. Figure 14 shows the visual input to the homing algorithm after image acquisition, unfolding and the application of the KLT tracker. The tracked features are superimposed on the image. It must be emphasized that although the homing experiment has been carried out in a single room, the



(a) (b)

Figure 12. Workspace layout for two representative homing experiments. Starting at point G, homing is achieved by visiting three and five MPs in experiments (a) and (b), respectively. The workspace dimensions are approximately 4×9 meters.

(a) (b) (c)

Figure 13. Snapshots from a homing experiment. The robot returns back to the home position after the execution of the prior path and the homing route with three MPs. These snapshots correspond to the experiment illustrated in Fig. 12(a).

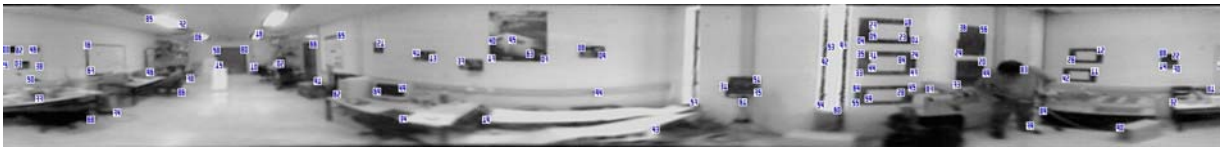


Figure 14. Cylindrical panoramic view of the workspace from the home position that the robot is approaching in Fig. 13. The features extracted and tracked at this image frame are also shown as numbered rectangles.

appearance of the environment at positions H and G differs considerably. As it can be observed, the robot has achieved the home position with high accuracy (the robot in Fig. 13(c) covers exactly the circular mark on the ground).

In a second experiment, a more complicated scenario was investigated. A number of panels were added to the workspace of Fig. 12(a), dividing it into two separate rooms communicating through a narrow passage (see Fig. 12(b)). Because of the introduction of the panels, the visual appearance of the two "rooms" is completely different, posing a challenge to feature tracking and to the process of defining the MPs. As it can be seen in Fig. 12(b), the robot defined five MPs in this experiment, as opposed to the three MPs of the first experiment. Figure 15 shows snapshots of the homing procedure. It can be verified that homing has been accomplished with an accuracy of a few centimeters.

(a) (b) (c)

Figure 15. Snapshots from the experiment in which the workspace has been separated into two rooms. The robot is visible (a) initially at its home position, (b) on its way back to the home position and (c) at the final position reached. This experiment corresponds to the workspace of Fig. 12(b).

6. Discussion

In this paper, a novel method for robot homing has been proposed. The method is based on tracking image corners in panoramic views of the environment. Tracking has been employed as a means to correspond features in different views of the environment. By memorizing and processing the "life-cycle" of the tracked corners, the robot is able to define MPs that should be revisited sequentially to achieve homing.

The proposed method has a number of attractive properties. A complex behavior such as homing is achieved by exploiting very simple sensory information. More specifically, only corners are extracted and tracked in a series of images and only the evolution of their retinal coordinates in a panoramic view is monitored. It is quite important that robot navigation, a problem typically handled through the use of range information, is achieved without computing any explicit range information. Argyros et al. (2004) present similar results for a reactive, corridor following behavior. The decomposition of the homing journey to a number of intermediate reactive navigation sessions appears intuitive. Moreover, the accuracy in the achievement of the home position does not depend on the distance traveled by the robot, which constitutes a fundamental problem in odometry-based homing. The final positional error depends only on the last step of the whole procedure (moving from the last MP to the home position).

This work also reveals the importance of omnidirectional visual cues to robot navigation tasks. The proposed scheme depends on the availability of a full 360° visual field in a number of ways. A robot equipped with a standard camera with limited visual field cannot easily identify "features seen before" on the way towards home position. Thus, homing becomes a much more difficult task. In environments that lack rich visual content, a panoramic sensor has higher probability of identifying features that are suitable for supporting navigation. A conventional camera would have much less candidate features to select and track because of its limited field of view. Furthermore, as shown earlier, the accuracy in reaching a certain position is improved when the employed features are distributed over a large field of view. With respect to the proposed method, an additional advantage of a 360° field of view is that it simplifies the description of the conditions under which the local control law is successful for all possible feature configurations.

An important issue in vision-based homing is the selection of the visual features that can support navigation. In our approach to homing, the selected features were image corners. The advantage of corners is that typically, there are many of them in most indoors environments. Their definition, extraction and tracking are mathematically rigorous and computationally efficient. Their main disadvantage is that correspondence of corners in views acquired from considerably different viewpoints can only be achieved through tracking. This is actually the reason why the proposed approach cannot easily be extended to more complex navigation tasks such as "go to location X". To support such a scenario would imply that the robot memorizes the life-cycle of the features in all paths connecting its current position to all potential target positions. To alleviate this problem an alternative would be to use, instead of corners, characteristic areas of the environment, such as visual landmarks. Current research work is towards this direction.

Notes

1. H is the initial position of the prior path, the home, and G the position where homing is initiated. S and T are any two positions along the path and they correspond to the starting and target positions for the local control law.
2. The term "catchment area" was originally proposed by Cartwright and Collett (1983).

References

Argyros, A.A., Tsakiris, D.P., and Groyer, C. 2004. Biomimetic centering behavior: Mobile robots with panoramic sensors. In *IEEE Robotics and Automation Magazine*, Special Issue on Panoramic Robotics, December 2004, pp. 21–30.

Baltzakis, H., Argyros, A.A., and Trahanias, P. 2003. Fusion of laser and visual data for reliable robot motion planning and collision avoidance. *International Journal of Machine Vision and Applications*, 15:92–100.

Basri, R., Rivlin, E., and Shimshoni, I. 1998. Visual homing: Surfing on the epipoles. In *the Proceedings of the Sixth International Conference on Computer Vision (ICCV-98)*, Bombay, India, pp. 863–869.

Bianco, G. and Zelinsky, A. 1999. Biologically inspired visual landmark learning and navigation for mobile robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99)*, Korea, pp. 671–676.

Burgard, W., Fox, D., and Thrun, S. 1997. Active mobile robot localization. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, San Mateo, CA.

Burgard, W., Trahanias, P., Haehnel, D., Moors, M., Schulz, D., Baltzakis, H., and Argyros, A.A. 2002. TOURBOT and WebFAIR: Web-operated mobile robots for tele-presence in populated

exhibitions. In *Proceedings of the IROS 02 Workshop on Robots* in Exhibition, EPFL, Lausanne, Switzerland.

Cartwright, B.A. and Collett, T.S. 1983. Landmark learning in bees: Experiments and models. *Journal of Computational Physiology*, 151:521–543.

Cartwright, B.A. and Collett, T.S. 1987. Landmark maps for honeybees. *Biological Cybernetics*, 57:85–93.

Cassinis, R., Grana, D. and Rizzi, A. 1996. A perception system for mobile robot localization. *Machine Learning and Perception, series in Machine Perception Artificial Intelligence*, Singapore, Vol. 23, pp. 57–64.

Chahl, J.S. and Srinivasan, M.V. 1997. Navigation, path planning and homing for autonomous mobile robots using panoramic visual sensors. In *the Proceedings of AISB Workshop on Spatial Reasoning in Mobile Robots and Animals*, Manchester, UK, pp. 47–55.

Choset, H. and Burdick, J. 2000. Sensor-based exploration: The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19:96–125.

Collett, T.S. 1996. Insect navigation en route to the goal: Multiple strategies for the use of landmarks. *The Journal of Experimental Biology*, 199:227–235.

Collett, T.S. and Rees, J.A. 1997. View-based navigation in hymenoptera: Multiple strategies of landmark guidance in the approach to a feeder. *Journal of Computational Physiology*, 181:47–58.

Cormen, T.H., Leiserson, C.E. and Rivest, R.L. 1996. *Introduction To Algorithms*. MIT Press, McGraw-Hill Book Company.

DeSouza, G.N. and Kak, A.C. 2002. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267.

Dyer, F.C. 1996. Spatial memory and navigation by honeybees on the scale of the foraging range. *Journal of Experimental Biology*, 99:147–154.

Facchinetti, C. and Hügli, H. 1994. Using and learning vision-based self-positioning for autonomous robot navigation. In the *Proceedings of the MLC-COLT Workshop on Robot Learning*, Rutgers University, New Brunswick, USA.

Fox, D., Burgard, W., Dellaert, F., and Thrun, S. 1999. Monte carlo localization: Efficient position estimation for mobile robots. In *the Proceedings of AAAI-99*.

Fox, D., Burgard, W., and Thrun, S. 1998. Active markov localization for mobile robots. *Robotics and Autonomous Systems*.

Franceschini, N., Pichon, J.M., and Blanes, C. 1992. From insect vision to robot vision. *Philosophical Transactions of the Royal Society of London*, 337:283–294.

Franz, M.O. and Mallot, H.A. 1998. Biomimetic robot navigation. *Technical Report No.65, Max-Planck-Institut für Biologische Kybernetik*.

Franz, M.O., Schölkopf, B., and Bülthoff, H.H. 1997. Homing by parameterized scene matching. *TR No.46, Max-Planck-Institut für biologische Kybernetik*.

Franz, M.O., Schölkopf, B., Mallot, H.A., and Bülthoff, H.H. 1998a. Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125.

Franz, M.O., Schölkopf, B., Mallot, H.A., and Bülthoff, H.H. 1998. Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, 79:191–202.

Gaussier, P., Joulain, C., Banquet, J.P., Leprtre, S., and Revel, A. 2000. The visual homing problem: An example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30(1/2):155–180.

Gutmann, J.S., Burgard, W., Fox, D., and Konolige, K. 1998. An experimental comparison of localization methods. In the *Proceedings of the 1998 IEEE/RSJ, International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada.

Kröse, B.J.A., Vlassis, N., and Bunschoten, R. 2002. Omnidirectional vision for appearance-based robot localization. In *Sensor Based Intelligent Robots: International Workshop, Dagstuhl Castle, Germany, October 2000*, G.D. Hagar, H.I. Cristensen, H. Bunke and R. Klein (Eds.), Selected Revised Papers, no 2238 Lecture Notes in Computer Science, Springer, pp. 39–50.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., and Wehner, R. 2000. Mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64.

Lourakis, M., Tzurbakis, S., Argyros, A.A., and Orphanoudakis, S. 2003. Feature transfer and matching in disparate views through the use of plane homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (T-PAMI), 25(2):271–276.

Matsumoto, Y., Sakai, K., Inaba, M., and Inoue, H. View-based approach to robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 3:1702–1708.

Möller, R. 2000. Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics, special issue in "Navigation in Biological and Artificial Systems"*, 83(3):231–243.

Rizzi, A., Duina, D., Inelli, S., and Cassinis, R. 2000. Unsupervised matching of visual landmarks for robotic homing using fourier-mellin transform. In *International Conference on Intelligent Autonomous Systems*, Venice, Italy.

Santos-Victor, J., Vassallo, R., and Schneebeli, H.J. 1999. Topological maps for visual navigation. In *the First International Conference on Computer Vision Systems*, Las Palmas, Canaries.

Shi, J. and Tomasi, C. 1993. Good features to track. Technical Report 93–1399, Department of Computer Science, Cornell University.

Srinivasan, M.V., Zhang, S.W., Lehrer, M., and Collett, T.S. 1996. Honeybee navigation en route to the goal: Visual flight control and odometry. *The Journal of Experimental Biology*, 199:237–244.

Thompson, S., Zelinsky, A., and Srinivasan, M.V. 1999. Automatic landmark selection for navigation with panoramic vision. In the *Proceedings of Australian Conference on Robotics and Automation ACRA'99*, Brisbane, Australia.

Thrun, S. 1999. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

Thrun, S. 2000. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. 2000. Robust monte carlo localization for mobile robots. *Artificial Intelligence*.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. 2000. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 101:99–141.

Tomasi, C. and Kanade, T. 1991. Detection and tracking of point features. *CMU-CS-91-132, School of Computer Science, Carnegie Mellon University*.

Trahanias, P., Burgard, W., Argyros, A.A., Haehnel, D., Baltzakis, H., Pfaff, P., and Stachniss, C. Tourbot and webFair: Web operated mobile robots for telepresence in populated exhibitions. To appear in *IEEE Robotics and Automation Magazine*, Special issue on EU-funded projects in Robotics.

Weber, K., Venkatesh, S., and Srinivasan, M.V. 1998. Insect inspired robot homing. *Adaptive Behaviour*.

Winters, N., Gaspar, J., Lacey, G., and Santos-Victor, J. 2000. Omni-directional vision for robot navigation. *IEEE Workshop on Omni-directional Vision (OMNIVIS'00)*, Hilton Head, South Carolina.

Winters, N. and Santos-Victor, J. 1999. Mobile robot navigation using omni-directional vision. In *the Proceedings of the 3rd Irish Machine Vision and Image Processing Conference (IMVIP'99)*, Dublin, Ireland.

Dr. Antonis A. Argyros is a research scientist at the Institute of Computer Science (ICS) of the Foundation for Research and Technology (FORTH) in Heraklion, Crete, Greece. He received his Ph.D. from the Department of Computer Science, University of Crete, Greece, in visual motion analysis. He has been a postdoctoral fellow at the Royal Institute of Technology in Stockholm, Sweden, where he worked on vision-based, reactive robot navigation. In 1999 he joined the Computational Vision and Robotics Laboratory of ICS-FORTH, where he has been involved in many RTD projects in image analysis, computational vision and robotics. Dr. Argyros has also served as a visiting associate professor at the Computer Science Department of the University of Crete. His current research interests include computational vision and robotics and particularly the visual perception of motion and 3D structure, the development of robot behaviors based on visual information and alternative visual sensors. In these research fields he has published more than 45 papers in peer reviewed scientific journals and conference proceedings.

Kostas Bekris is a Ph.D. candidate in Computer Science at Rice University. He is a member of the Physical and Biological Computing Group under the the supervision of Prof. Lydia Kavraki. His research interests include mobile robot navigation with sensing constraints, motion planning and robotic sensor networks. Kostas Bekris received his B.S. in Computer Science at University of Crete, Greece in 2001 and an M.Sc. in Computer Science from Rice in 2004.

Professor Stelios C. Orphanoudakis holds a Ph.D. degree in Electrical Engineering from the Thayer School of Engineering, Dartmouth College, USA, a M.S. degree in Electrical Engineering from the Massachusetts Institute of Technology (MIT), and a B.A. degree, magna cum laude with highest distinction in Engineering Sciences, from Dartmouth College. He held a faculty appointment in the Departments of Diagnostic Radiology and Electrical Engineering at Yale University, USA, from 1975 until 1991. Since 1986, he holds a faculty appointment as Professor of Computer Science at the University of Crete, Greece. Furthermore, from 1991 until 1994, he was Acting Director of the Institute of Computer Science-FORTH (ICS-FORTH) and, from 1994 until 2004, he was Director of this Institute. Today he is Chairman of the Board of Directors of FORTH, and he is scientific leader of the Center for Medical Informatics and Health Telematics Applications and of the Computational Vision and Robotics Laboratory. Prof. Orphanoudakis is a member of many honorary and professional societies and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). He has many years of academic and research experience in the fields of computational vision and robotics, intelligent image management and retrieval by content, medical informatics, and medical imaging. He is the author of more than 120 publications in international scientific journals, refereed conference proceedings and books. He has served on various committees and working groups of the European Commission and has been active in numerous European R & D programs. He has also served on the Board of Directors of the EuroPACS society (1994–2000). During the period 1995–2000, he served on the National Telecommunications and Post Commission of Greece. Finally, from 1994 until 2001, he served on the National Advisory Research Council of Greece and, from 1998 until 2002, he served on the Board of Directors of the Hellenic Foundation for Culture.

Lydia E. Kavraki is the Noah Harding Professor of Computer Science and Bioengineering at Rice University. Kavraki received a B.A. degree in Computer Science from the University of Crete in Greece, and a Ph.D. in Computer Science from Stanford University. Her

research interests are in physical algorithms and their applications in robotics, computational structural biology, and bioinformatics. Kavraki is one of the principal inventors the Probabilistic RoadMap planner (PRM). Her research interests include the development of methods for robot planning in high dimensions and with physical constraints, assembly planning, micromanipulation using microelectromechanical systems, and flexible object manipulation. In her recent work, Kavraki applies robotics methods to the modeling of biomolecular interactions for drug design. Kavraki is the recipient of the 2000 Association for Computing Machinery (ACM) Grace Murray Hopper Award for her technical contributions. She has also received an NSF CAREER award, a Sloan Fellowship, and the 2002 Early Academic Career Award from the IEEE Society on Robotics and Automation. Kavraki has recently been included in the list of Top 100 Young Innovators under 35 of the MIT Technology Review Magazine and in the list of Top 10 Young Scientists by the Popular Science Magazine. She is a member of IEEE and ACM and a Fellow of AIMBE.