

Robot Homing based on Corner Tracking in a Sequence of Panoramic Images

Antonis A. Argyros, Kostas E. Bekris and Stelios C. Orphanoudakis

Computer Vision and Robotics Laboratory, Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
Heraklion, Crete, Greece
Email: {argyros, bekris, orphanou}@ics.forth.gr

Abstract

In robotics, homing can be defined as that behavior which enables a robot to return to its initial (home) position, after traveling a certain distance along an arbitrary path. Odometry has traditionally been used for the implementation of such a behavior, but it has been shown to be an unreliable source of information. In this work, a novel method for visual homing is proposed, based on a panoramic camera. As the robot departs from its initial position, it tracks characteristic features of the environment (corners). As soon as homing is activated, the robot selects intermediate target positions on the original path. These intermediate positions (IPs) are then visited sequentially, until the home position is reached. For the robot to move between two consecutive IPs, it is only required to establish correspondence among at least three corners. This correspondence is obtained through a feature tracking mechanism. The proposed homing scheme is based on the extraction of very low-level sensory information, namely the bearing angles of corners, and has been implemented on a robotic platform. Experimental results show that the proposed scheme achieves homing with a remarkable accuracy, which is not affected by the distance traveled by the robot.

1. Introduction

The goal of this research is to study the behavior of visually guided homing. Homing is a term borrowed from biology [1], where it is usually used to describe the ability of various living organisms, such as insects, to return to their nest after having traveled a long distance along a certain path. Similarly, robots may be required to return to their initial position after moving around in their

environment in order to perform a certain task. If this is achieved using visual information, the behavior is known as visually guided homing.

There have been many efforts to solve the problem of autonomous robot navigation, incorporating a homing behavior, using non-visual sensors. Robots are usually equipped with odometry sensors that continuously measure the location of the robot with respect to some relative or absolute coordinate system. Unfortunately, the errors of these sensors are considerable and most importantly, cumulative. The longer and more complex the path of the robot is, the larger the errors in position estimation become. For this reason, odometry is used as an indicator of the robot's position, which is supplemented by information provided by ultrasound or laser sensors. When such sensors are used, navigation and localization is usually achieved through the creation of a grid-based map of the environment [2]. Although many of these methods are quite successful, grid-based approaches imply a high space and time complexity that depends on the number of grid-cells. Moreover, robots based solely on proximity sensors get easily confused because the sensors often return similar readings at different locations. Many efforts have been made to solve the problem of uncertainty in robot localization for proximity sensors [3,4].

By comparison, vision is much more informative, since it provides information regarding “where is what”. Nature provides many examples of visual systems that are successful in supporting long-range navigation. For example, insects such as bees and ants exhibit remarkable homing abilities, although their neural system is relatively small and not sophisticated. Cartwright and Collett [5] have proposed the so-called snapshot model for insect navigation. They suggest that ants and bees fix the locations of landmarks surrounding a position by storing a snapshot of the landmarks taken from that position. This snapshot represents a constellation of objects and it does not encode explicitly any information regarding 3D structure. Another interesting finding [6,7], is that insects

* This work was partially supported by the EU-IST project TOURBOT – Contract No IST-1999-12643.

correct their path en-route to the target position by executing biased detours according to familiar landmarks, which act as beacons.

The models of insect behavior described so far, provide interesting ideas on how to solve the homing problem for autonomous robots. The use of vision and, especially, the exploitation of a large visual field have proven to be essential for the accomplishment of various navigational tasks in insects. In addition, the selection of certain parts of images as areas of interest and their use as landmarks plays also a major role in the way biological systems deal with visual data. Last, but not least, computed information regarding structure does not seem to be a prerequisite for navigation.

Many efforts at achieving visual homing in robotics have taken into consideration the models from insect physiology that has already been described. Lambrinos et al [8] implemented directly the model of Cartwright and Collett on the Sahabot II. The experiments were conducted on a flat plane in the Sahara desert with four black cylinders used as highly distinctive landmarks. The snapshots were aligned with a compass direction, obtained from the polarized light compass of Sahabot. Franz et al [9] analyze the computational foundations of snapshot-based guidance and provide proofs on convergence and error properties based on a proposed alternative model to that of Cartwright and Collett. In [10], Möller describes the average landmark vector technique, which is closely related to the snapshot model, and its implementation on analog hardware. Cassinis et al [11] have also dealt with bee-inspired visual navigational models. In [12], a neural architecture is proposed which uses panoramic visual cues, detects landmarks and aims at navigating a robot. In [13], a method is proposed for the selection of landmarks that can support robot navigation. A common characteristic of all these approaches is that they exploit the benefits of the wide field of a panoramic observer, so as to obtain more information about the environment.

The work presented in this paper exploits omnidirectional vision. The proposed method relies on the fact that, if there is a correspondence established among at least three features in two panoramic views taken from two different positions, then there is a control algorithm that can drive the robot from one position to the other. In this work, the extracted features are corners, which can easily be defined, extracted and localized in images. To establish corner correspondence between views acquired from considerably different viewpoints, the KLT corner tracker [14] has been employed.

The proposed method has been implemented and extensively tested on a robotic platform. The accuracy of the homing procedure is quite high; home position is achieved with an accuracy of a few millimeters, after a

robot journey of approximately 10 meters, during which the robot performed a number of maneuvers.

The rest of this paper is organized as follows. In section 2, the algorithmic steps of the proposed method for visual homing are described in more detail. Section 3 presents representative results from extensive experiments that have been carried out using a robotic platform. Finally, the proposed method is further discussed in section 4.

2. Method Description

The method for robot homing proceeds as follows. The robot starts moving at a certain position (home position). As it departs from this position, it extracts characteristic features (corners) and tracks them in subsequent image frames. As the robot moves, some of the corners disappear, while some others appear. The system keeps a record of the “life-cycle” of all detected corners. At a certain moment in time, it is decided that homing should be activated. Then, the algorithm computes a number of IPs on the original path. The basic idea behind this selection process is to guarantee that between two consecutive IPs, a sufficient number of features survive (theoretically a minimum of three), so that the robot can move between these two positions in a vision based, reactive manner. In practice, the number of features that should survive is larger in order to guarantee a robust performance. The rest of this section describes the above algorithmic scheme in more detail.

2.1. Moving between adjacent positions

The basic building block of the proposed approach is a method that drives a robot between two positions S and G , provided that there is an established correspondence among three features in the panoramic views acquired from these positions.



Figure 1: Sample panoramic image

A panoramic image can easily be unfolded using a polar-to-Cartesian transformation [15]. Figure 1 shows an example of a panoramic image and Fig. 6(a) - 6(j) contains examples of unfolded ones. The property of the resulting image is that the full 360 degrees field of view is mapped on its horizontal dimension. In the remainder of this paper, unless otherwise stated, the term panoramic image refers to an unfolded panoramic image.

Let F_1, F_2 and F_3 denote three features of the environment. Let also S and G denote the panoramic images acquired at start S and goal G positions, and S_i, G_i be the bearing angles of features F_i in images S and G , respectively. Since we deal with panoramic images, the bearing angle of a feature can easily be computed as:

$$S_i = \frac{2\pi x_i}{S_{size}}, \quad (1)$$

where x_i is the x-coordinate on the feature in the image and S_{size} is the width of the panoramic image (measured in pixels). Two features F_i and F_j define two sectors in the images acquired at positions S and G . The angular sizes of these sectors are $S_{ij} = |S_j - S_i|$ and $G_{ij} = |G_j - G_i|$, respectively.

We define a control strategy that can move the robot between two adjacent positions S and G , by exploiting only the bearing angles of three features. The method assumes that features F_i can be corresponded between images S and G . Given a retinotopic ordering of the features, we define consecutive sectors in images S and G . Since features can be corresponded, sectors can also be corresponded. For each sector we record the difference in its angular extend between images S and G . If the difference is positive, we define a vector parallel to the bisector of this sector with a magnitude proportional to the difference $|G_{ij} - S_{ij}|$. If the difference is negative, then the vector has opposite direction. In fact, each of these vectors is targeted towards the direction of the motion that is required in order to equalize the angular extend of the sector between the two views. The vector sum of all contributions from all sectors defines a vector M (see Fig. 2).

M is not necessarily directed exactly towards target position G . It has been experimentally validated that if this control law is applied continuously, then the robot converges to the goal position G . Reaching the target position can easily be verified by checking whether the magnitude of the motion vector M is lower than a predefined threshold.

The method requires monitoring of at least two sectors and, consequently, the correspondence of at least three features between two views. With less than three features the control law fails to move the robot to the goal

position. If more than three features can be corresponded between different views, then all of them can be used to define the global motion vector M . Simulations have shown that as the number of features increases, vector M becomes parallel to the line defined by S and G , with a direction towards G . This is also the case if the features are uniformly distributed around position G .

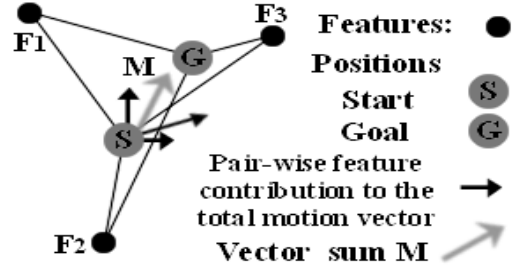


Figure 2: Graphical presentation of the method used for moving between adjacent positions based on the bearing angle of features.

Feature correspondence may introduce errors in the estimation of the bearing angle of features. One way to eliminate some of these errors is to exploit the ordering of features, as they appear in panoramas acquired in 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 vector M . In order to detect these features, the Longest Common Sub-string Algorithm [16] that is based on dynamic programming, has been employed. Formally, given two sequences $X = \langle x_1, x_2, \dots, x_k \rangle$ and $Z = \langle z_1, z_2, \dots, z_m \rangle$, Z is considered a subsequence of X if there exists a strictly increasing sequence $\langle i_1, i_2, \dots, i_k \rangle$ of indices of X such that, for all $j = 1, 2, \dots, k$, $x_{i(j)} = z_j$. 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 the sequence of features $FS = \langle F_1, F_2, \dots, F_M \rangle$ in the start position and the sequence $FG = \langle F_1, F_2, \dots, F_K \rangle$ in the goal position. The estimation of the motion vector M (from S to G) is based on the features contained in the maximum-length common subsequence of FS and FG . The time complexity of the Common Sub-string Algorithm is $O(M+K)$.

An important aspect of this control strategy is that it achieves the goal position G , but does not necessarily fix the pose of the robot. This is because it employs information regarding the difference of the bearing angles and not the absolute values of the bearing angles themselves. However, once the target position is achieved, pose correction is an easy task because panoramas acquired from the same position but with different pose are shifted versions of the same image.

The proposed method drives a robot from a starting position \mathbf{S} to a nearby position \mathbf{G} by exploiting the bearing angles of features that implicitly contain structure information. Compared to the original snapshot model proposed by Cartwright and Collett, no external compass is required to align views. Moreover, it avoids the definition of centrifugal motion vectors.

2.2. Extracting features from images

The control law for moving between adjacent positions is based on the availability of feature correspondences between two panoramic views. If, however, the two views have been acquired from a significantly different viewpoint, feature correspondence is not a trivial task [17]. For this reason, the developed homing behavior achieves feature correspondence through feature tracking in a series of panoramic images that the robot acquires as it moves. This guarantees small inter-frame displacements, which, in turn, facilitates the task of feature correspondence. More specifically, the KLT tracking algorithm [14] has been employed. KLT starts by identifying characteristic image features (corners), which then tracks in a series of images. An important property of KLT is that the definition of features to be tracked is exactly the one that guarantees optimal tracking. The definition of corners is based on the quantity

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

which is defined over an $N \times N$ neighborhood of image point (i,j) . In eq. (2) g_x and g_y are the partial derivatives of the image intensity function. The eigenvalues λ_1 and λ_2 of the Z matrix are computed. Good features to track are considered those satisfying the rule

$$\min \{ \lambda_1, \lambda_2 \} > t \quad (3)$$

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

$$e = \sum \sum [I - J] \begin{bmatrix} g_x \\ g_y \end{bmatrix} \quad (4)$$

where I and J are the two images containing the features to be tracked and g_x, g_y are the gradients of the image intensity function. The minimization of e is based on the solution of the linear system

$$Zd = e \quad (5)$$

where d is the displacement of the tracked feature. Except the purely translational model, tracking can take into account the case of an affine transformation between two consecutive images. Theoretically, the second model is more general and allows tracking of features that have undergone shearing or rotation. Practically, it leads to poor results and many false matches. Shi and Tomasi [14] propose the use of the translation model for a good displacement measurement of features and confine the affine model to monitor a feature's quality by checking the dissimilarity between the initial and the current frame.

2.3. Definition of intermediate positions during homing

Based on the KLT feature selection and tracking algorithm, the robot defines corners in the view acquired at its home position. As it departs from this position it continuously tracks these corners in the frames it acquires. During its course, some of the initially selected features may not satisfy the criterion of eq. (3) while other features may appear that satisfy this criterion. In the first case the system “drops” the features from subsequent tracking. In the second case, features are being tracked. The system builds some type of internal memory in which information regarding the “life-cycle” of features is kept. A graphical presentation of this type of memory is shown in Fig. 3.

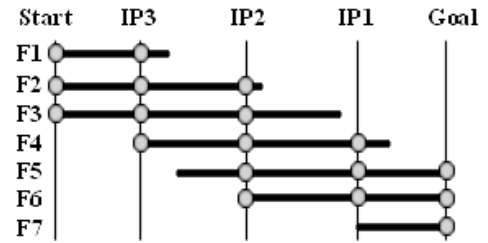


Figure 3: The life-cycle of features tracked between the initial position and the goal position.

The vertical axis in this figure corresponds to the features that have been identified and tracked during the journey of the robot from home position to goal position. The horizontal dimension corresponds to time. Horizontal lines correspond to the life cycle of a certain feature. In this particular example, the start and the goal position do not share any common feature and, therefore, the simple control law presented in section 2.1 cannot be employed to directly support homing. In order to alleviate this

problem, the notion of intermediate target positions is introduced.

Being at the goal position, the algorithm decides how far the robot can go towards the home position. In terms of the control law described in section 2.1, this position is the one that is closer to the home position and shares at least three common features with the goal position. For the example of figure 3, this position is denoted as “IP1”. Achieving “IP1” from the end position is, by definition, feasible by employing the control strategy of section 2.1. The algorithm proceeds in a similar manner to define the next IP towards home. The procedure terminates when the last IP achieved coincides to the home position.

It should be noted that the control strategy of section 2.1 guarantees the achievement of the target position but not necessarily the achievement of the pose with which the robot has previously visited this position. This poses a problem in the process of switching from the features that drove the robot in a certain IP to the features that will drive the robot to the next IP. To solve this problem, we exploit the fact that panoramic images that are acquired at the same location from a different pose are shifted versions of the same image. Moreover, the amount D (in pixels) of shift is proportional to the difference in pose, ϕ . Theoretically, one feature suffices for the computation of ϕ , because features are tracked and their bearing angle in both images is known. Practically, all available features should contribute to get a robust estimation of ϕ . Errors can be due to the inability of KLT to track accurately some features and due to the non-perfect achievement of an IP. In our implementation, ϕ is calculated as the median of angular shifts suggested by all tracked points.

Having an estimation of the displacement ϕ between the images taken in an IP during the initial and the homing path, it is possible to restart a new homing procedure. The features detected during the first visit at this IP are recovered and their coordinates are transformed according to the displacement ϕ . Feature selection is then applied to small windows centered at the predicted locations. The result of this calculation is considered as the current position of the old feature in the image.

An important implementation decision is the selection of the number of features that should be corresponded between two consecutive IPs. In section 2.1 it has been shown that three features suffice, although more features can be used if available. It is quite important that the minimum number of required features is only three. Provided that in both indoor and outdoor environments corners exist in abundance, this almost guarantees that there will be a series of IPs that when sequentially visited, lead the robot to its home position. The advantage of considering more than three correspondences is the fact that the achievement of IPs

(and consequently the achievement of the home position) becomes more accurate because tracking errors are smoothed out. On the other hand, if the number of correspondences is unnecessarily too high, this can lead to the definition of an unnecessarily large number of IPs.

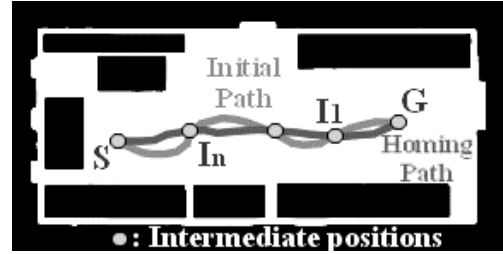


Figure 4: Graphical representation of the method used for exploiting IPs so as to return to home after a long journey in space. S and G are the initial and final robot positions. The robot returns to home position S by sequentially visiting IPs I_1, I_2, \dots, I_n .

Another important aspect of the proposed homing method is that 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 error does not accumulate and the achievement of the home position depends only on the last step of the whole procedure (moving from the last IP to the home position).

3. Experimental results

The proposed method has been implemented on a real robotic platform equipped with a panoramic camera. A Pentium III processor running at 750 Mhz consisted the processing power used for both vision processing and robot motion control. Testing took place in a real office environment where several experiments have been conducted. Due to space limitations, sample results are presented. Figure 4 gives an approximate layout of the robots’ workspace and the robots’ start position. This position was marked on the ground to facilitate the evaluation of the homing procedure. The robot was driven towards the empty space covering a distance of approximately ten meters. During this motion, panoramic images were acquired and processed on line. The robot’s maximum translational speed was 4.0 cm/sec and its maximum rotational velocity was 3 deg/sec.

After the execution of the initial path, three intermediate target positions were selected so as to guarantee that at least 80 features would be constantly available during homing. Figure 6 shows snapshots of the homing experiment. More specifically, Fig. 5(a)-5(h) correspond to the initial robot journey and Fig 5(h)-(o)

correspond to the homing procedure. The robot has reached the home position with sub-centimeter accuracy (the mark on the ground is covered by the robot in both figures 5(a) and 5(o)). For comparison, the odometry-based homing resulted in an error of approximately 1.5 meters for exactly the same homing task.

Figures 6(a)-6(e) show samples of the panoramic views that the robot acquired during its initial journey and figures 6(f)-6(j) show samples of the panoramic views that the robot acquired during homing. In both figures the corners extracted and tracked are also displayed. From these views it can easily be verified that although the homing experiment has been carried out in one room, the appearance of the environment from the home and end positions differs considerably.

4. Discussion

In this paper, a novel method of visual has been proposed. The method is based on the tracking of corners in panoramic views of the environment. By memorizing and processing the “lifecycles” of such corners, the robot is able to define IPs that should be visited sequentially before reaching the home position.

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 position on a panoramic view is monitored. It is quite important to note that robot navigation, typically handled by exploiting range information, is achieved without computing any explicit range information at all. The decomposition of the homing journey to a number of intermediate reactive navigation sessions appears quite intuitive. Moreover, the fact that, between IPs, the robot moves in a reactive manner enables the robot to accommodate its behavior according to changes in the environment such as obstacles etc, at least to the point that these changes do not affect substantially the overall appearance of the environment.

This work also reveals the importance of omnidirectional vision in robot navigation tasks. In environments that lack rich visual content, a panoramic sensor has higher probability of capturing features that can support navigation. A standard camera would have much less candidate features to select and track because of its limited field of view. Furthermore, the experiments show that homing accuracy is improved when the robot exploits information in 360 degrees.

In our approach to visual homing, the selected features were corners. Current research efforts are targeted towards exploiting more complex visual features.

References

- [1] K. Weber, S. Venkatesh, M. V. Srinivasan, “Insect Inspired Robot Homing”, *Adaptive Behaviour*, 1998.
- [2] S. Thrun, “Learning Metric-Topological Maps for Indoor Mobile Robot Navigation”, *AI Journal*, 99(1), 21-71, 1999
- [3] D. Fox, W. Burgard, S. Thrun, “Active Markov Localization for Mobile Robots”, *Robotics and Autonomous Systems*, 1998.
- [4] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo Localization for Mobile Robots” *AI Journal*, 2000.
- [5] B.A. Cartwright, T.S. Collett, “Landmark Maps for Honeybees”, *Biol. Cybernetics*, 57, 85-93, 1987.
- [6] T.S. Collett, “Insect Navigation *en Route* to the Goal: Multiple Strategies for the Use of Landmarks”, *Journal of Exper. Biology*, 199, 227-235, 1996.
- [7] M.V. Srinivasan, S.W. Zhang, M. Lehrer, T.S. Collett, “Honeybee Navigation *en route* to the Goal: Visual Flight Control and Odometry”, *The Journal of Exper. Biology*, 199, 237-244, 1996.
- [8] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, R. Wehner, “A Mobile Robot Employing Insect Strategies for Navigation”, *Robotics and Autonomous Systems*. 30, 39-64, 2000.
- [9] M. Franz, B. Schölkopf, H. Mallot, H. Bülthoff, “Where Did I Take That Snapshot? Scene-based Homing by Image Matching”. *Biological Cybernetics*, 79, 191-202, 1998.
- [10] R. Möller, “Insect Visual Homing Strategies in a Robot with Analog Processing” *Biological Cybernetics*, special issue: *Navigation in Biological and Artificial Systems*, Vol. 83, No. 3, 231-243, 2000.
- [11] R. Cassinis, A. Rizzi, G. Bianco, N. Adami, P. Mosna, “A Biologically-Inspired Visual Homing Method for Robots”, *Workshop AIIA-IAPR-IC*, Ferrara, 1998.
- [12] P. Gaussier, C. Joulain, S. Zrehen, J-P. Banquet, “Visual Navigation in an Open Environment Without a Map”, *Proc. IROS 97*, Grenoble, 1997.
- [13] S. Thompson, A. Zelinsky, M.V. Srinivasan, “Automatic Landmark Selection for Navigation with Panoramic Vision”, *Proc. of Australian Conf. on Robotics and Automation ACRA'99*, Brisbane, Australia, March 1999.
- [14] J. Shi, C. Tomasi, “Good Features to Track”, TR 93-1399 Department of Computer Science, Cornell University, November 1993.
- [15] D. Tsakiris, A.A. Argyros, “Corridor Following by Nonholonomic Mobile Robots Equipped with Panoramic Cameras”, IEEE MCCA'00 Inv. Session, "Sensor-based Control for Robotic Systems", Patras, July 2000.
- [16] T.H. Cormen, C.E. Leiserson, R.L. Rivest. “Introduction To Algorithms”, MIT Press, McGraw-Hill Book Company, 1996.
- [17] M. Lourakis, S. Tzurbakis, A. Argyros, S. Orphanoudakis, “Using Geometric Constraints for Matching Disparate Stereo Views of 3D Scenes Containing Planes”, ICPR'00. Oct. 2000, Barcelona, Spain.

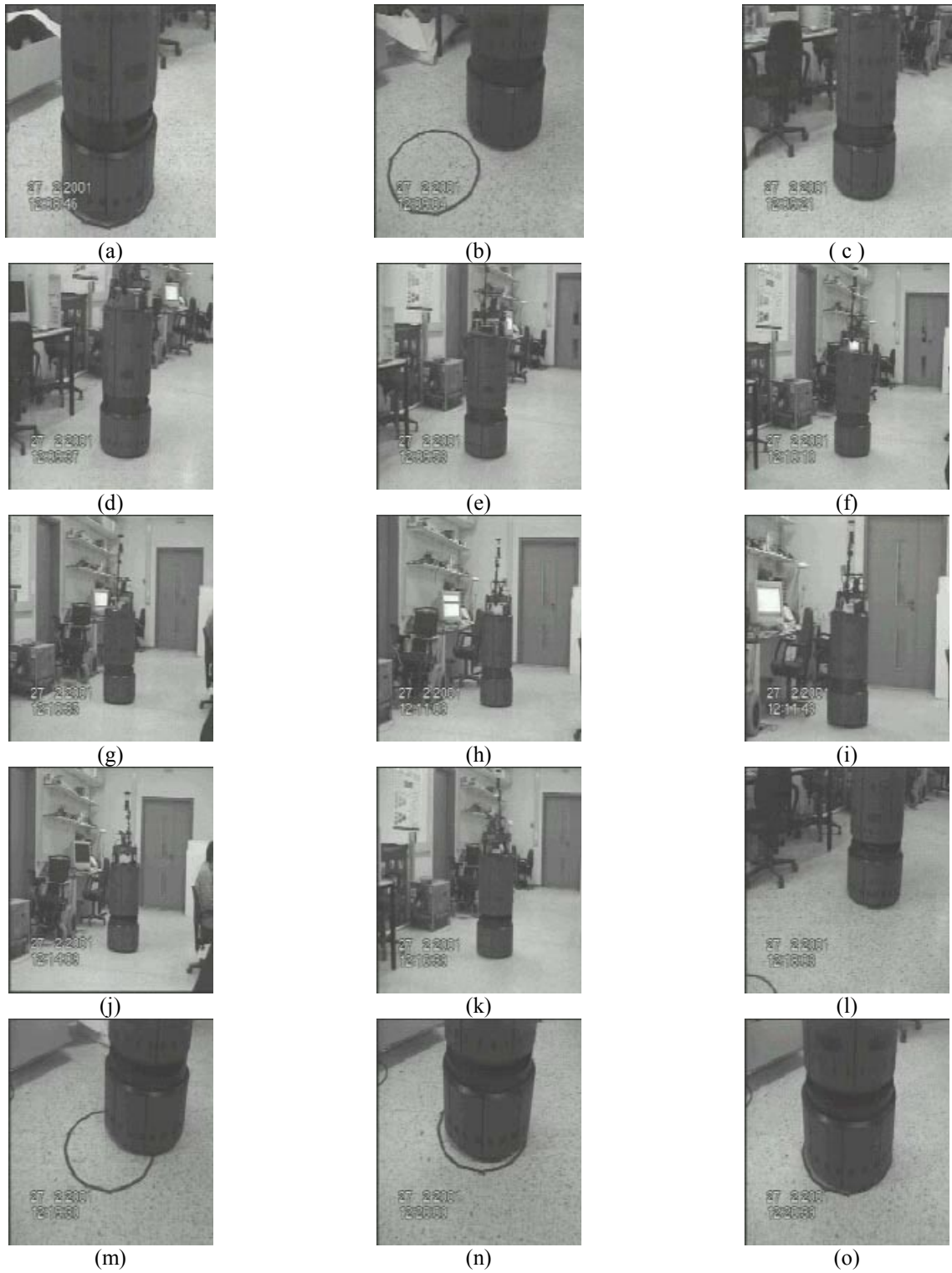


Figure 5: Views of the robot while homing: Robot departs from designated home position ((a) to (h)) and then it returns back ((h) to (o))



Figure 6: Panoramic images and corner tracking as the robot departs from and returns back to the home position