Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Image-Based Robot Navigation in Unknown Indoor Environments

Ehud Rivlin[†] , Ilan Shimshoni[‡] , Evgeny Smolyar[†]
[†]Dept. of Computer Science, [‡]Dept. of Ind. Eng. and Mgmt.
Technion - Israel Institute of Technology
32000 Haifa, Israel

***Abstract*— This paper presents a method for image based robot navigation under the full perspective model. The robot navigates through unknown indoor environments. A target image is taken from an unconstrained position in the environment and given to the robot. The robot starts at an arbitrary position and navigates to the position at which the target image was taken. The approach is based on using images of the environment taken by the robot at different positions along the path and comparing them with a target image. No extraction of 3D models of the scene is needed. The robot finds automatically an image which shows part of the environment shown in the target image. It then moves on the floor, takes pictures with its camera, finds corresponding features in the current and target image, and uses them to extract the motion parameters to the target location. All these steps are performed automatically. This paper describes experimental results performed with a Nomad XR4000 mobile robot. These experiments show the feasibility and the significant benefits of our approach.**

I. Introduction

Various algorithms in the field of vision-based navigation of autonomous indoor vehicles were developed in recent years [9], [18]. Indeed, in an industrial environment, the presence of autonomous mobile robots performing repetitive tasks is more and more predominant. Moving a robot to the desired position and orientation in such an environment is significant particularly in the context of a flexible environment. This task is usually performed by providing the robot with the model of the environment [21], [22], [19]. The main drawback of this technique in the context of flexible environments is its rigidity. Human intervention is needed in order to redefine the task as the environment changes. The second drawback of this approach is the loss of accuracy due to measurement error accumulation. Our system can be attributed to the relatively new View-Based approach, which utilizes the appearance of the scene [7], [20], [15] and does not require map building. Consequently, it is much more flexible. Still, the huge memory requirements and computational cost associated with these algorithms constitute their major disadvantage. In [17], [16], for example, the sequence of frontal views along the route has to be memorized which is particularly problematic if the starting position of the robot may vary.

This article describes a vision based navigation algorithm for a mobile robot, which constitutes a further investigation of the paradigm described in [1], [2]. Related work in the general field of visual servoing has also been performed in [14], [24], [3], [13]. The target pose is specified only by an image taken from that pose. This is the only input the robot gets. As the robot moves along its path, it takes pictures that together with the target image are used to estimate the direction of motion to the goal. In addition the robot is able to autonomously find the route to a desired object which appears in the target image, even when this object changes its position. In this paper the problem of a real mobile robot moving in indoor environment with three degrees of freedom is considered. No previous knowledge of the environment is necessary. One of the most important contributions of this project is the fully automated robust implementation of the algorithm. No human intervention is required at any stage of the navigation.

The paper is structured as follows. In Section II we present an overview of the approach. Section III introduces the theoretical basis of the image-based robot navigation under the full perspective projection model. Section IV discusses the robust implementation of the algorithm. In Section V the results of the experiments performed with a Nomad XR4000 mobile platform are presented. We summarize the paper and discuss future research directions in Section VI.

II. Overview

The framework in which the proposed navigation algorithm works is as follows. A mobile robot is taken to a small number of positions in an indoor environment. At each target position an image is taken by the robot's onboard camera. The target positions and orientations are selected such that at each position in the environment there exists an orientation from which part of the scene shown in one of the target images is visible. At runtime the robot is placed in an unknown position in the environment. It then looks for the initial orientation for which there exists an overlap in scenes. This enables it to start the algorithm. Thus initially, it rotates on its place taking significantly overlapping pictures. Then, from the array of these pictures, it automatically finds the one that overlaps with the target image. This image is considered as the

0-7803-7860-1/03/$17.00 © 2003 IEEE

starting image. This way, no constraints on the initial position are imposed.

In the second part of the algorithm the current and target images are compared. Features are extracted from both images and matched trying to find correspondences. The quality of these matches are usually very low. Even so the epipolar geometry relating the two images is estimated. From the recovered epipolar geometry the relative orientation of the two imaging positions and the direction to the target position is estimated. The robot starts moving towards the goal taking additional images on the way. These images are used to recover the distance to the goal and improve the quality of the estimate of the target position. Since the estimate is repeated at each image, localization errors do not accumulate. The block diagram of the algorithm is shown in Fig. 1.

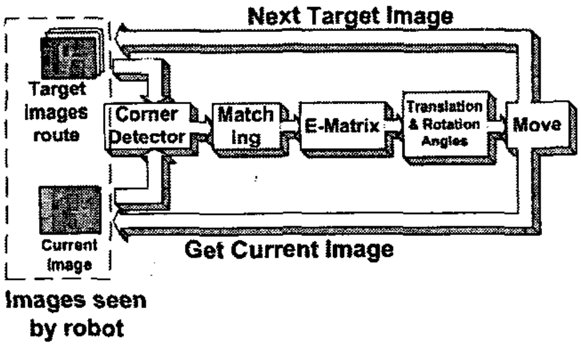


Fig. 1. The block diagram of the multi-target navigation algorithm.

Two navigation strategies are possible. In the first strategy after the parameters for translation and rotation needed to arrive to the target have been estimated, the robot moves a small fraction of the distance to the goal. Then it takes another image. Before continuing in the proposed direction, it checks that it is now closer to the target than before the movement and updates its parameters. This process repeats until the goal is achieved. The second strategy works iteratively. After calculating the translation and rotation parameters, the robot moves to the estimated goal. Then it takes another image and treats it as the initial image, calculating the parameters again. The iteration process continues until the difference between the current position image and the target image is small enough. The second algorithm is faster requiring less images than the first algorithm. It is however less robust because errors in the estimation can yield images which do not overlap with the target scene which will require to repeat the first step of the algorithm which is time consuming.

Under this paradigm the robot does not have to move directly to the goal. It can pas on the way through some intermediate targets. Consequently, the robot can navigate through these landmarks towards targets that it cannot see from the initial position. For example, the robot will be capable of finding targets that are situated in other rooms. Another use of intermediate targets is to enable the robot to pass through some tight spots such as doors by taking an intermediate target image at the door.

In the following sections we will describe in detail the components of the algorithm and its implementation.

III. Full- Perspective Robot Navigation

In this section, we consider the problem of image-based robot navigation when the full-perspective projection model is assumed. Our goal is to move the robot to an unknown target position and orientation S, which is given in the form of an image I of the scene taken from that position. At any given step of the algorithm the robot is allowed to take an image I' of the scene and use it to determine the next move. Denote the current unknown position of the robot by S', our goal then is to lead the robot to S.

A. Extrinsic Camera parameters recovery

Before running the algorithm the camera used in the experiments is calibrated and its intrinsic parameters represented by the matrix K are recovered. We consider a camera that is rigidly positioned on the robot. The extrinsic parameters of the image with respect to the target image uniquely determine the position and orientation of the robot. To determine the motion of the robot we have to recover the relative position and orientation S' of the robot relative to the target pose S from the corresponding images I' and I. By finding sufficiently many correspondences in the two images we can recover information about the motion parameters relating the two poses. These are the translation vector $t = (t_x, 0, t_z)$ which can be recovered only up to scaling factor along with the rotation angle θ. From the recovered information about the translation , we can deduce the direction to the target position. In order to obtain an estimate for the distance to the target we need an additional image which is taken by robot as it moves towards the goal.

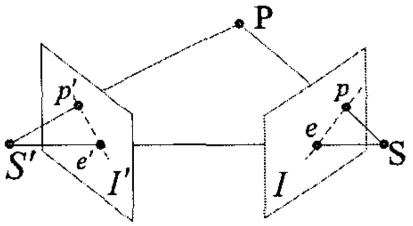


Fig. 2. Epipolar Geometry.

B. Epipolar geometry

Epipolar geometry describes a geometric relationship between the positions of corresponding points in two images. We examine the case where the intrinsic calibration parameters of the cameras have been estimated in advance. In our case the difference in the positions of the two cameras is due to motion in the plane parallel to the floor (the $X \circ Z$ plane) and rotation about the Y axis.

Let p and p' be the projections of a 3D point P on the target and current image respectively. The calibrated points are: $\hat{p} = K^{-1}p$ and $\hat{p}' = K^{-1}p'$, where K is the calibration matrix. Since the robot can rotate only about the Y-axis, the rotation matrix is:

$$R = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix}.$$

Let the translation of the robot in the plane $X \circ Z$ be $t = (t_x, 0, t_z)^T$. Thus the epipoles in the calibrated images are $e = (t_x/t_z, 0, 1)$ and $e' = Re$. The epipolar constraint relating corresponding points can be expressed as:

$$\hat{p}_i'^T Rt \times \hat{p}_i = 0. \tag{1}$$

Here × denotes the vector product. Introducing a skew-symmetric matrix

$$[t]_\times = \begin{pmatrix} 0 & -t_z & 0 \\ t_z & 0 & -t_x \\ 0 & t_x & 0 \end{pmatrix}$$

This constraint (1) can be rewritten in matrix form as :

$$\hat{p}_i'^T E \hat{p}_i = 0, \tag{2}$$

where $E = R[t]_x$ is called the essential Matrix. This constraint limits the position of the corresponding point of $\hat{p}_i$ $\hat{p}_i'$ to lie on an epipolar line $l' = E\hat{p}_i$ which goes through the epipole e' and for $\hat{p}_i$ to lie on an epipolar line $l = E^T\hat{p}_i'$ which goes through e.

In our case, the essential matrix is of the type:

$$E = \begin{pmatrix} 0 & -\cos(\theta) + \sin(\theta)\frac{t_x}{t_z} & 0 \\ 1 & 0 & -\frac{t_x}{t_z} \\ 0 & \sin(\theta) + \cos(\theta)\frac{t_x}{t_z} & 0 \end{pmatrix}. \tag{3}$$

In order to recover E by solving equation (2) linearly, three correspondences are needed. This is a variation on the general eight-point algorithm [11], [5], [6]. The parameters $e_x = t_x/t_z$ and θ can be recovered from the essential matrix (3).

In order to deal with the uncalibrated points in the images we substitute for $\hat{p}_i', \hat{p}_i$ in (2) yielding:

$$p_i'^T K^{-T} E K^{-1} p_i = 0.$$

We define the fundamental matrix F such that

$$p_i'^T F p_i = 0. \tag{4}$$

This matrix is related to the essential matrix by

$$E = K^T F K. \tag{5}$$

In order to solve the equation (4) linearly, eight correspondences are needed which constitute a major drawback compared to the method that takes advantage of the special simplified structure of E in our this case which requires only three matches.

C. Finding the distance to the target

As the rotation angle and the moving direction have been recovered from two images, the only unknown parameter that remains is the distance to the target which will require another image. After acquiring the first image computing the motion direction and rotation angle, the robot starts to move in the proposed direction. It moves a relatively small distance and acquires a second image. Now we have three images - the images from the first, second and target positions. Moreover, we have an estimate for the distance traveled between the first and the second position, as we know how far the robot moved. The obtained images are denoted I, I', I'', where I is the target image and I' and I'' are the current and the previous images respectively. The robot made a step of size αt and the remaining number of steps of size αt to the target position is $\nu = 1/\alpha$ which have to be recovered.

First, we compensate for the rotation between the poses of the images by applying the inverse rotation matrices to the points recovered from the images. Now the image planes are parallel. Thus the image poses are separated only by translation in the direction of $t = (t_x, 0, t_z)$. The epipoles of all images are at $e = (t_x/t_z, 0, 1)$. Given a point in the scene $P = (X, Y, Z)$ the x coordinates of the calibrated projections of P to the three images are:

$$x = \frac{X}{Z}, x' = \frac{X + t_x}{Z + t_z}, x'' = \frac{X + (1+\alpha)t_x}{Z + (1+\alpha)t_z} \tag{6}$$

As shown in [2], by eliminating X and Z the following expression can be received:

$$\nu = \frac{1}{\alpha} = \frac{(x' - x)(x'' - e_x)}{(x'' - x')(x - e_x)} \tag{7}$$

The x coordinates can be replaced by the position along the epipolar line. So ν is actually the cross ratio along the epipolar line. The geometric interpretation of (7) could be seen in Fig. 3.

ν indicates not only the distance but the direction as well. A positive ν will indicate that the robot is moving towards the target position where as a negative ν will mean that it is moving away from the target pose. Theoretically, only one point triplet is enough to recover ν, but in

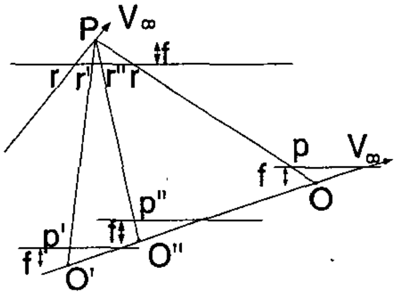


Fig. 3. The Geometric interpretation of the cross-ratio.

practice, it is desirable to examine all the points in order to retrieve a reliable estimate, since the error can be quite significant for a single triplet.

IV. Implementation

The main building block of our system is a module which recovers the essential matrix E from a pair of calibrated images. First, the SUSAN corner detector [23] is applied to both images. Then corresponding points are found based on local image similarity. The problem is however that many of the proposed matches are incorrect as can be seen in (Fig. 4). We will therefore have to employ a robust technique to recover the essential matrix.

As the degrees of freedom of the motion of the robot are known, they yield constraints on the essential matrix. Two ways to calculate the essential matrix exploiting these constraints are explored. The first way is to estimate linearly the fundamental matrix first from 8 matching pairs of points, disregarding at this point the special structure of the essential matrix. Next the intrinsic calibration matrix K is applied to it to compute the essential matrix. In the final stage the closest essential matrix of type (3) is found and the motion parameters t_x/t_z and θ are extracted from it. The second method is to calculate an estimate for an essential matrix of the type we are searching from the start from a match of three points only. This method is better because it requires less matches for the estimation stage and because it yields an essential matrix which satisfies the special constraints of our case from the beginning.

A. RANSAC algorithm

The last problem that we have to deal with is the large number of incorrect matches. To solve this problem we apply the RANSAC paradigm [4]. The main steps of this algorithm for essential matrix calculation are as follows:

◇ Repeat for N samples:

- Select a random sample of m correspondences from the initial set and compute E (3). In our case $m = 3$.
- Compute from E and K the fundamental matrix F.
- Calculate the distance d_i for each putative correspondence p_i, p'_i from its corresponding epipolar lines l and l' respectively, where

$$\begin{aligned} d_i &= dist(p_i, l_i)^2 + dist(p'_i, l'_i)^2 \\ &= (p'^T_i F p_i)^2 \\ &\quad \left(\frac{1}{(Fp_i)^2_x + (Fp_i)^2_y} + \frac{1}{(F^T p'_i)^2_x + (F^T p'_i)^2_y} \right) \end{aligned} \tag{8}$$

- Compute:

$$d = \sum_{i=1}^{n} \min(d_i, \delta) \tag{9}$$

a score for the current hypothesized E, where δ is a preset threshold estimating the maximal value a d_i of an inlier pair can get for a correct E with high probability.

◇ Choose the E with the smallest value of d.

The number of trials N has to chosen to ensure with high probability p that at least one of samples of m selected pairs is free from outliers. Following [10], N is chosen to be at least as high as

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^m)},$$

where ϵ is the outlier proportion in the initial set.

By examining this equation it is obvious in the first method which we considered in which $m = 8$, a much larger number of trials would be needed to find a correct match then in the second method. In addition in cases where the number of correct matches is small we might not have enough inliers at all to compute the essential matrix and verify its correctness.

V. Experimental Results

The algorithms described above have been implemented and real world experiments were conducted in the lab on a Nomad XR4000 robot. The robot was equipped with a Canon VC-C3 Camera . The camera has been calibrated based on the method suggested by [25].

At the beginning, the robot was placed at two target positions and target images were taken. The XR4000 was then placed at an arbitrary position and orientation in the environment.

In the first stage, the robot rotates about its central axis by 360° taking a picture every 18° and comparing it with the target image (Fig. 5(a)). For every current image-target image pair, the essential matrix and its score d is computed using the procedure described above. The image whose essential matrix yields the lowest score is chosen. In the refinement stage, the robot rotates ±5° around the position found in the first stage with a 1° step

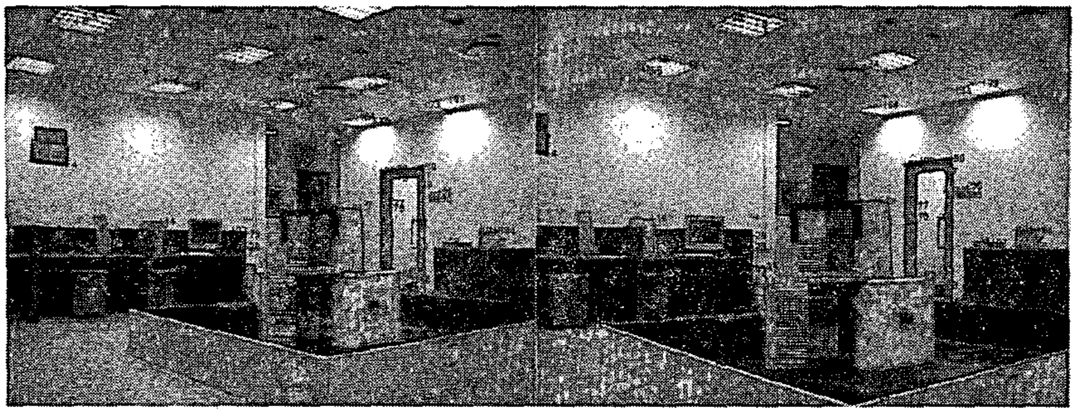


Fig. 4. Two images of the same scene are presented. The View captured by the robot from different positions. The extracted corresponding points have the same numbers on the images. Not all the corresponding points are found correctly.

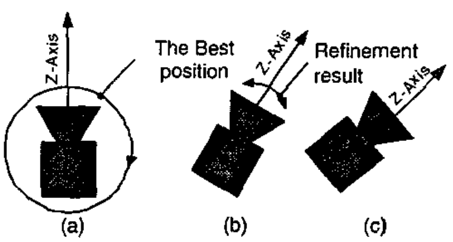


Fig. 5. Finding the start position for the navigation algorithm. First 360° rotation (a), second refinement around the best position (b) and finally, positioning the robot on the position calculated from the second stage (c).

(Fig. 5(b)). Fig. 6 demonstrates the experimental results of the initial orientation finding stage showing for each image the number of point pairs whose distance d_i was less than δ and are therefore assumed to be correct.

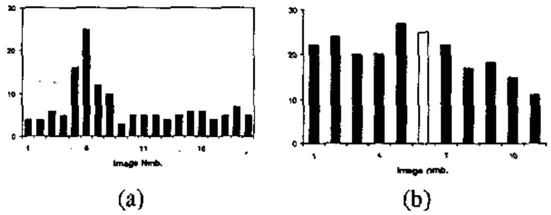


Fig. 6. Number of correctly associated corresponding points vs. image number: The results of initial orientation (a) and refinement stage (b). Here the column that demonstrates the optimal result, found from the initial orientation, is white.

After the most suitable image is found, it begins the image-based navigation algorithm image (Fig. 5(c)).

The results of two experiments, which constitute a two target path, are considered below. The approximate environment maps are presented in Fig. 7(a)-(c). In these figures, the hatched regions exhibit the fields of view from different points of view. When the hatched regions of two positions overlap, the shared region can be seen from both positions.

In the first experiment the robot navigates from the starting position 7(d) to the first target position 7(e). The hatched region 7(a) shows that there is a large overlapping region between the fields of view of the start and the target position. This means that the navigation algorithm can be applied. Fig's 9(a)-(f) show the image sequence taken by camera on the robot in the first experiment as well as the robot position sequence. The robot successfully navigates from the start position 9(a) to the target position 9(g).

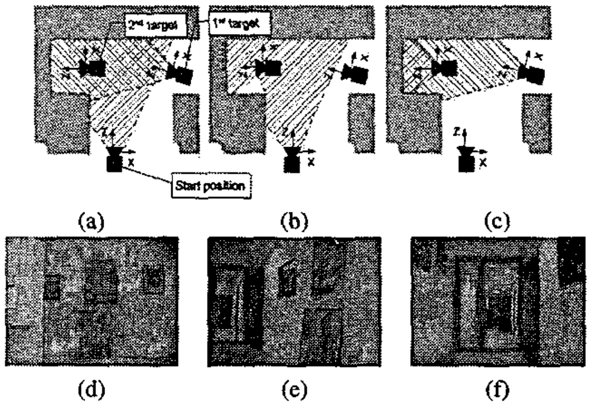
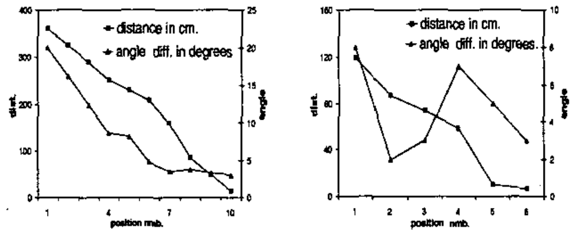


Fig. 7. The Multi-Target case. (a)-(c): the map of the experiment environment. The view from the initial position (d), the 1^{st} target position (e) and the 2^{nd} target position (f).

In the second experiment the robot navigates from the start position 7(d) to the first target position 7(f). The robot could not navigate directly to the second target position from the initial position (7(d)) since the second target image scene (7(f)) could not be seen from the start position (Fig. 7(b)). However, the robot can navigate to the second target position from the first target position (Fig. 7(c)) and to the first target position from the start position (Fig. 7(a)). A route that consists of two targets has been defined. The first target position is identical to the one of the first experiment. The second target is demonstrated in Fig. 7(f). The results of this experiment are shown in Fig. 9(f)-(n). The graphs of the robot position and orientation relative to the target's position and orientation for these two experiments are shown in Fig. 8.



(a) (b)

Fig. 8. Experimental results: The difference between the robot's current position and orientation and the target's pose at every step for (a)first and (b) second part of the experiment.

VI. Conclusions

In this paper we have described a new algorithm for image-based robot navigation. The implementation developed in the framework of this project is robust and does not require human intervention at any navigation stage. Our approach does not require either a predetermined model of the environment nor information about the 3-D position of the robot. The main idea is to retrieve the essential matrix linking the image taken by the camera mounted on the robot with the target image. The parameters estimating the translation direction and the rotation between the current robot position and the target robot position can be retrieved from the essential matrix. An additional image is required to estimate the distance to the goal.

The system is highly efficient since it only stores three images, namely the current image, the previous image and the target image, at any given time. Another advantage of this algorithm is that it is not restricted to a predetermined planar path. Once it receives the target image, the robot determines its path on-line. Finally, applying the multi-target algorithm, one can make the robot navigate to a target that was not initially in its field of view.

In this paper, we have ignored the issue of possible obstacles on the path of the robot. However, this issue can be solved by combining of the navigation algorithm described in this article with an obstacle avoidance algorithm such as the Bug algorithms [12], [8].

There are a couple of issues that remain to be addressed. First, the navigation algorithm can be broadened to deal with more than three degrees of freedom as well as with the possibility of onboard camera movement and rotation. This could be extremely important in industrial environments, where robots with degrees of freedom higher than three are common.

Second, the system can be adapted to work in flexible environments. That means that the robot will be trained to find a path to the desired position relative to some object even though the object itself can change its location. For example, the navigation goal could be to find a cup in the room, although the cup itself can be on an arbitrary place on the table or under the table.

Finally, the target images can be used as a flexible map of the environment where at each point in time the robot can know its position with respect to one or more of the target images and be given commands to reach any place in the environment and not only one of the target positions.

Acknowledgments

The support of the Israeli Ministry of Science under the grant 2104 is gratefully acknowledged.

VII. References

[1] R. Basri and E. Rivlin. Localization and homing using combinations of model views. *AI*, 78(1-2):327–354, October 1995.

[2] R. Basri, E. Rivlin, and I. Shimshoni. Image-based robot navigation under the perspective model. In *IEEE Int. Conf. on Robotics and Automation*, pages 2578–2583, 1999.

[3] C. Collewet, F. Chaumette, and P. Loisel. Image-based visual servoing on planar objects of unknown shape. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 247–252, Seoul, Korea, May 2001.

[4] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.

[5] R. Hartley and A. Zisserman. *Multiple Views Geoemtry in Computer Vision*. Cambridge University Press, 2000.

[6] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(6):580–593, June 1997.

[7] S.D. Jones, C. Andresen, and J.L. Crowley. Appearance based processes for visual navigation. *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pages 551–557, Sept. 1997.

[8] I. Kamon and E. Rivlin. Sensory-based motion planning with global proofs. *IEEE Trans. on Robotics and Automation*, 13(6):814–822, December 1997.

[9] A. Kosaka and A.C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Computer Vision, Graphics, and Image Processing-Image Understanding*, 56(3):271–329, 1992.

[10] A. Lacey, N. Pinitkarn, and N. Thacker. An evaluation of the performance of RANSAC algorithms for stereo camera calibrarion. *BMVC*, 2000.

[11] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

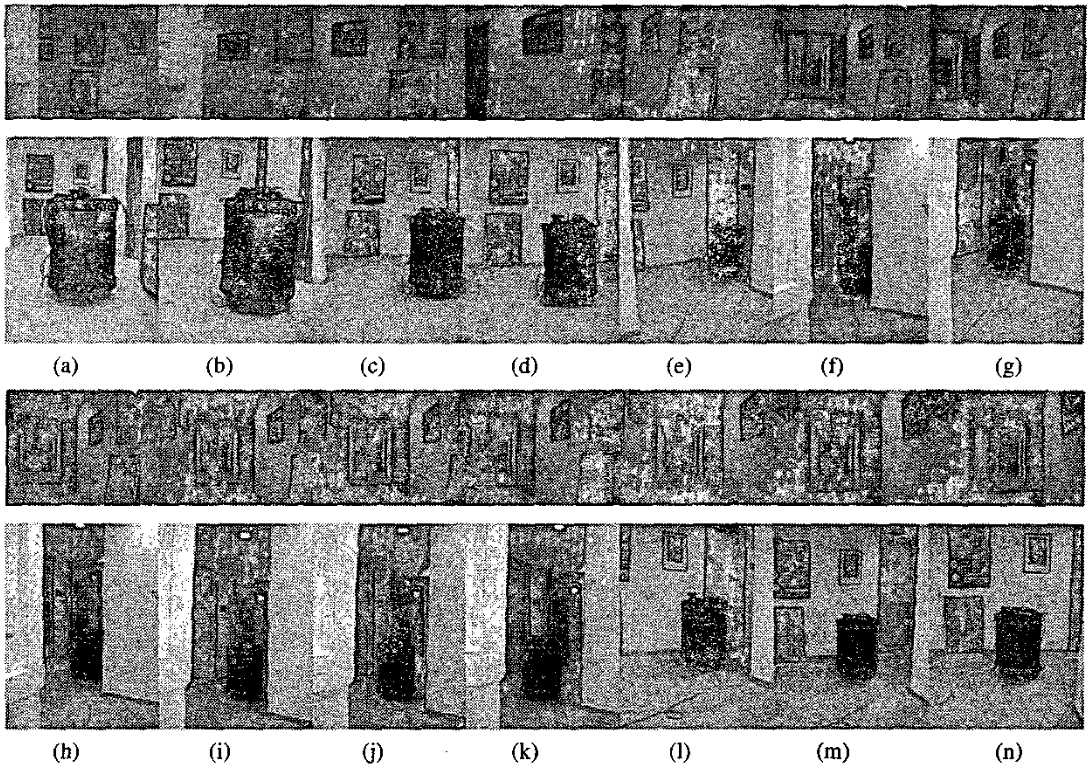


Fig. 9. Multi-Target Session snapshots. On the top: Image sequence taken by the camera on the robot. On the bottom the position of the robot (a) the starting point (b)-(e) snapshots from transitional positions. (f,h) Final position for the 1st target (g) 1st target. (i)-(l) snapshots from transitional positions towards the 2nd target, (m) the final position (n) the 2nd target.

[12] V. Lumelsky and T. Skewis. Incorporating range sensing in the robot navigation function. *IEEE transactions on systems, man and cybernetics*, 20(5):1058–1069, 1990.

[13] R. Mahony, T. Hamel, and F. Chaumette. A decoupled image space approach to visual servo control of a robotic manipulator. In *IEEE Int. Conf. on Robotics and Automation, ICRA'02*, volume 3, pages 3781–3786, Washington DC, May 2002.

[14] E. Malis, F. Chaumette, and S. Boudet. 2-1/2-D visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, April 1999.

[15] A. Martynez and J. Vitria. Clustering in image space for place recognition and visual annotations for human-robot interaction. *IEEE Trans. Systems, Man, and Cybernetics*, 31(5), Oct. 2001.

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

[17] Y. Matsutmoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *Proceedings of IROS99*, pages 317–322, 1999.

[18] M. Meng and A.C. Kak. Neuro-nav: A neural network based architecture for vision-guided mobile robot navigation using non-metrical models of the environment. *Proc. IEEE Int'l Conf. Robotics and Automation*, 2:750–757, 1993.

[19] H.P. Moravec and A. Elfes. High resolution maps from wide-angle sonar. In *IEEE Int. Conf. on Robotics and Automation*, pages 1151–6, April 1986.

[20] T. Ohno, A. Ohya, and S. Yuta. Autonomous navigation for mobile robots referring pre-recorded image sequence. *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, 2:672–679, Nov. 1996.

[21] K. Onoguchi, M. Watanabe, M. Okamoto, Y. Kuno, and H. Asada. A visual navigation system using a multi-information local map. In *IEEE International Conference on Robotics and Automation*, pages 767–774, 1990.

[22] K.B. Sarachik. *Visual Navigation: Constructing and Utilizing Simple Maps of and Indoor Environment*. MIT Press, March 1989.

[23] S.M. Smith and J.M. Brady. SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78, May 1997.

[24] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch. Matching of affinely invariant regions for visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, pages 1601–1606, 1999.

[25] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Proc. Int. Conf. Comp. Vision*, pages 666–673, September 1999.