A Navigation System with Self-Localization Capabilities for an Autonomous Mobile Robot

Alain Bétourné, Alessandro Fefè, , Salvatore Nicosia, Paolo Valigi

Dipartimento di Informatica, Sistemi e Produzione
Università di Roma "Tor Vergata" – 00133 – Roma – Italy
{nicosia,valigi}@disp.utovrm.it

Abstract

In this paper a navigation system for an autonomous mobile robot is presented, based on the joint use of an odometry, encoder based sensory system, of a vision system, and of an ultrasonic captor sensory system. A flexible hierarchical data structure is used to model complex indoor environments. The navigation system plans and controls robot movements, and performs periodic corrections of the robot position, by means of a vision system. Experimental tests have been performed, showing the good accuracy achieved by the proposed position correction scheme.

1 Introduction

The problem of navigation and motion planning for mobile robot is a subject of great interest, and over the past years it has received considerable attention (see, among others, [1, 2, 3] and references therein). Several problems arise in the design of navigation systems for mobile robots to be operated in real scenarios. The problem of dynamically updating the available knowledge/map of the environment, and the problem of determining with a sufficient degree of accuracy the robot position, are among the most crucial ones.

The need for updating the knowledge/map of the environment arises in all the situations in which the environment changes with time, or in which the environment is only partially known to the robot.

As for the problem of position estimation, a simple odometry, encoder based, sensory system, usually available on all the platforms, while simple to use and cheap, is not fully satisfactory due to accumulation errors, arising, e.g., from wheel slipping.

Vision based sensory systems are very useful in dealing with these type of problems, since they allow to collect a large amount of information, in real-time. Also, a widely used approach is that of sensor fusion, by which an overall reduction in the ambiguities in the information can be achieved, as well as an increase in the performance of the sensory system.

In this paper, a navigation system is presented, comprising a flexible data structure modeling the environment, and a sensory system based on the joint use of encoder based odometry, ultrasonic captors, and artificial vision system. The algorithm proposed for image analysis, is based on the use of the Hough transform. Experimental results are presented, carried out on a LABMATE mobile platform.

Other navigation systems have have been presented in [4, 5, 6]. In [4] standard marks are placed in known locations, in order to allow the adjustment of position information. In [5] the robot navigation system is based on position estimation carried out by using incremental encoder data, fused together with an inertial navigation system. In [6] the "edge visibility regions" approach is used, in order to determine the robot position. The use of Hough transform allows to select a number of regions where the robot may be in. Then, by a searching method, the most likely robot position is identified.

2 Problem formulation

The general task afforded by a navigation system for an Autonomous Mobile Robot (AMR) is that of driving the robot from a given initial position, to a final destination, possibly performing specific duties while in movement. The basic element needed for planning a trajectory that could allow the robot to reach the destination, is a map describing the environment (see Figure 1). In general, in indorr environments, the path from the origin to the destination will go through several rooms, hence the whole path can be split into a set of simpler segments (see Figure 1):

- room crossing (i.e., a door-to-door path);
- door crossing (i.e., a room-to-room path).

The planning problem for a door-to-door segment can be solved by means of several algorithms (see, e.g., [3]). The planning problem for the room-to-room segment do not pose major problems as such, but it requires accurate information on current robot position. In this paper, the problem of indoor path planning and control for mobile robot is considered. Special attention is devoted to the execution of the room-to-room path segment, and to the associated position estimation problem.

3 Environment modeling

The navigation of an AMR requires the availability of a model of the environment in which it has to operate. The approach used to model the environment affects the computational efficiency of the whole system. Here, a hierarchical model is considered, which allows to use information in a selective manner. In particular, only the information describing the hierarchical levels required for the fulfillment of the current task can be used, thus achieving a considerable saving in computation time. The considered model is a modified version of the one introduced

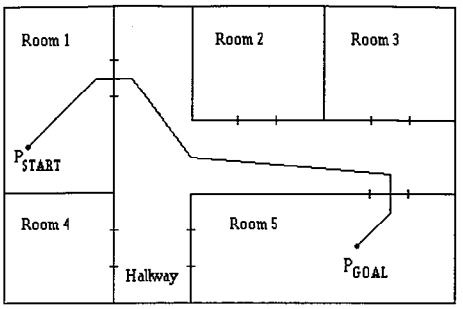

Figure 1: The environment map, and the path

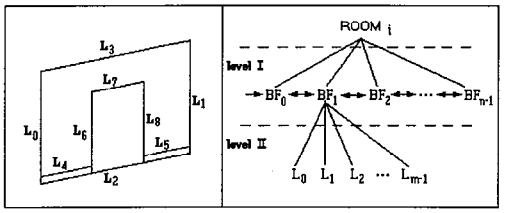

Figure 2: The hierarchy: description of a room

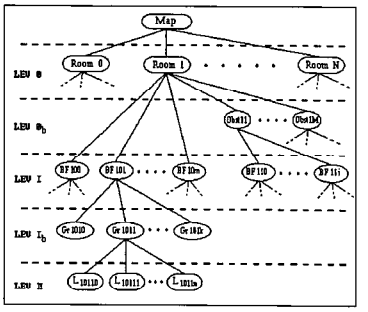

Figure 3: The whole hierarchy

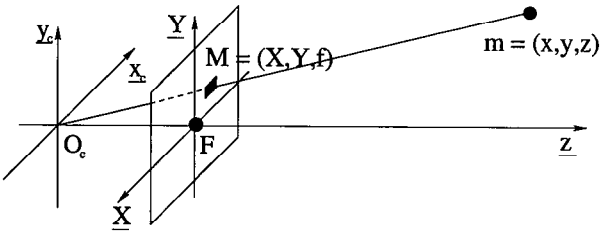

Figure 4: Camera model

in [7]. While simple, it contains all the information that a rendering algorithm would need to construct expectation maps for all the interesting cases. In the following, it is assumed that a reference frame $R_e = (O_e, x_e, y_e, z_e)$ is associated to the environment, with the plane (x_e, y_e) being the horizontal plane.

Fundamental for the considered environment model is the notion of *basic face*. It is assumed that a basic face is a vertical planar entity of unspecified height bounded by two vertical lines. In the proposed model, a face is represented by the 3-ple: $(BF_j, (x_1, y_1), (x_2, y_2))$, where BF_j, is the symbolic name of the basic face, and the other two entries are the world coordinates of the line formed by the projection of the basic face onto the horizontal (x_e, y_e)-plane. The set of all the basic faces corresponds to the first level of the hierarchy (see Figure 2).

A basic face points to all the lines that are deemed by a human to be significant from the standpoint of scene interpretation. For example, the basic face in Figure 2 will point to the 9 lines shown in the same figure. Each line of a basic face is represented by the 4-tuple $(L_i, BF_j, (x_1, y_1, z_1), (x_2, y_2, z_2))$ where L_i is the symbolic name of the i-th line, BF_j is a pointer to the face which contains the line, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the coordinates of the two end points of the line. A complete room may therefore be represented by the tree data structure shown in Figure 2.

Such a structure, which is the basic scheme proposed in [7], represents the model of one room alone; then, the map of a larger environment will simply comprise a set of rooms. Taking into account the problems considered in this paper, the above model has been extended. In particular, here doors have a particular importance, therefore, an auxiliary structure, called *group* (of lines), has been introduced. The purpose of this structure is to better manage all the lines comprising a basic face: the first group, *group 0*, will collect all the lines that do not belong to any door, the other groups collect the lines corresponding to the doors (the first door in group 1, the second door in group 2, and so on). Hence, the new tree data structure has two more levels, *level* 0, describing rooms, and *level* I_b, describing doors through groups (see Figure 3). For the purpose of this paper, an additional hierarchical level, *level* 0_b, in between the room level and the basic face level, is used, to describe obstacles. A picture of the whole hierarchy is reported in Figure 3.

4 Vision system

The camera-based vision system comprises a camera, installed on the mobile robot platform LABMATE, available at the Robotics and Industrial Automation Laboratory of the University of Rome "Tor Vergata", and an image processing software package, partially developed locally. The camera allows to obtain images of the scene with a resolution of 512 × 512 pixels in 256 grey levels, and is attached to the robot so that the camera axis is perpendicular to the robot wheel axis. An ideal model of the camera is considered, with a geometry of perspective projection type (See Figure 4).

Let $R_c = (O_c, \underline{x}_c, \underline{y}_c, \underline{z}_c)$ be a reference frame attached to the camera, and let $\underline{z}_c$ be the camera axis. The focal distance of the camera, i.e., the distance between the camera frame origin O_c and the camera focus point F, is denoted by f. The point $m = (x, y, z)$, $x, y, z, \in \mathbb{R}$, on the scene is projected onto the image plane on the point $M = (X_u, Y_u, f)$, $X_u, Y_u, f \in \mathbb{R}$, and the coordinates of

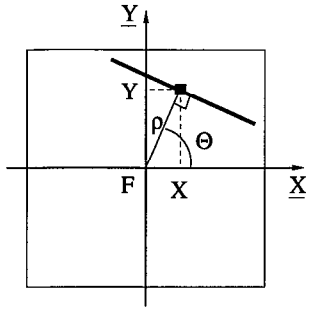

Figure 5: Parametrization of a line

the two points are related by the following equations:

$$X_u = \frac{f}{z}x, \quad Y_u = \frac{f}{z}y. \tag{1}$$

The absolute coordinates X_u, Y_u in (1) can be transformed into pixel coordinates X and Y, $(X, Y \in \mathbb{Z})$, by simply multiplying (1) by two scalar constants $-k_X$ and k_Y. Hence, equations (1) become:

$$X = -\frac{f_X}{z}x, \quad Y = \frac{f_Y}{z}y \tag{2}$$

where $f_X = fk_X$, and $f_Y = fk_Y$.

A *planar primitive* of the image (e.g., a curve) is denoted by $g(X, Y, \underline{q}) = 0$, where $\underline{q}$ is the configuration vector of the primitive, and X, Y are the pixel coordinates of a generic point belonging to the primitive. For example, if the primitive is a line, the vector $\underline{q}$ is given by $\underline{q} = \underline{q}_\ell$, with $\underline{q}_\ell = (\Theta, \rho)$, $\Theta, \rho \in \mathbb{R}$, where Θ and ρ are the spherical coordinates of the line with respect to the frame attached to the image (See Figure 5); $\Theta \in [0, \pi)$, and $\rho \in \mathbb{R}$. In the image plane, the line is represented by the equation:

$$g(X, Y, \underline{q}_\ell) = 0, \tag{3}$$

which can be written as:

$$X \cos\Theta - Y \sin\Theta - \rho = 0. \tag{4}$$

A spatial primitive $g_s(\mathbf{X}, \mathbf{q}) = 0$ can be introduced in a similar manner, with the vector $\mathbf{X}$ representing the coordinate of a generic point on the primitive, and $\mathbf{q}$ representing the primitive configuration vector [8].

5 Image transformation

Several techniques for image segmentations have been analyzed, and the one based on the two basic steps of image filtering and image segmentation has been chosen.

Image filtering

The images are processed by means of a filter based on the Laplace method and on the use of pixel intensity value threshold. The output of such a filter is an image with only two grey levels, where the intensity level is either 0 or 255, the value 255 corresponding to a pixel with an high contrast with the surrounding pixels. The contrast intensity level for a pixel in position (X, Y) will be denoted by $c(X, Y)$.

Segmentation

As the image resulting by the above filter has not $2D$ primitives with regular thickness, the filtered image is transformed into an image with regular thickness edges. by means of the Hough transform [9, 10, 11].

The (Θ, ρ) line parametrization shown above is used. In order to reduce the complexity of the line segmentation procedure, a decomposition approach has been chosen. The full size, 512 × 512 pixels, image (the *original image* or *level-1 image*), is divided into 4 sub-images (*level-2 images*), each level-2 image is again divided into 4 *level-3 images*, each level-3 image is divided into 4 *level-4 images*, and finally each level-4 image is divided into 4 32 × 32 pixel images, referred to as *elementary images* or level-5 images. Thus, the original image comprises 256 elementary images. The Hough transform allows to determine, for each elementary image, the lines corresponding to all the contained edges; then, the data characterizing the lines in all the elementary images are merged together, and all the lines of the original image are progressively reconstructed.

Analysis of a single elementary image. The set of line orientation $[0, \pi)$ is divided into intervals of equal length $\Delta\Theta$ equal to $arctg(\frac{1}{31})$ (the minimum value of Θ is $arctg(\frac{1}{31})$), and the set of line orientations is divided into $n_\Theta = \text{int}(\frac{\pi}{\Delta\Theta}) = 97$ intervals.

For each point of an elementary image of coordinates (X, Y) having a contrast value $c(X, Y) = 255$, and for a particular choice $\Theta = \Theta_i$ of the orientation angle, the value $\rho_i := X \cos\Theta_i + Y \sin\Theta_i$ is computed, and the number n_i of pixels corresponding to the same value $\rho = \rho_i$ is determined. For the purpose of comparison, two values of ρ have been considered equal if their difference is smaller than a tolerance threshold, chosen equal to 0.5 in this paper. If n_i is greater than or equal to a given threshold (chosen equal to 10 in the experiments presented here), then the line (Θ_i, ρ_i) represents a possible edge.

Once all the pixels with contrast value equal to 255 have been considered, it has to be verified whether two lines can be merged, being part of the same edge. If the difference between two values ρ_1 and ρ_2 is less than or equal to a threshold (chosen as 1.0 in this paper), a unique mean value is computed as $\rho = \frac{n_1\rho_1 + n_2\rho_2}{n_1 + n_2}$ where n_1 and n_2 are the number of pixels corresponding to the lines identified by ρ_1 and ρ_2, respectively, and by the same value of the orientation angle Θ.

Now, denote with $C = (\Theta, \rho, n)$ the sub-class corresponding to the possible edge having orientation angle Θ, distance ρ and approximating n pixels. All the subclasses can be arranged in ascending order with respect to the value of Θ. If only a finite number of known lines are of interest, e.g., in view of the specific problem under consideration, it is computationally more efficient to restrict the attention to a set of orientation angles close

to the ones of these lines.

Merging of results. Once all the line parameters have been calculated, for all the elementary images, then they are merged by means of the following procedure.

1. Set $k = 5$ for the first time, else let $k = k - 1$.
2. The elementary images are regrouped 4 by 4 to constitute a larger level-$(k-1)$ image.
3. The spherical coordinates (Θ_i, ρ_i) of each level-k image k are transformed into the reference frame of the new level-$(k-1)$ image. The number of sub-classes of this new image is equal to the sum of the sub-classes of each level-k image.
4. The parameters of all the possible lines of the level-k images are compared themselves and merged according to Algorithm 1, where *NbSubClasses* and *NbLines* denote the number of sub-classes and the number of possible lines, respectively.
5. Steps 1 to 4 are repeated until an image having the same size as the original one is obtained.

Algorithm 1 *Edge Detection Algorithm*

$$\left|\begin{array}{lcl} NbLines & = & 1 \\ j & = & 1 \\ merging & = & 0 \end{array}\right.$$

for $i = 1..NbSubClasses$

  while $(j <= NbLines)$ or $(merging = 0)$

    if $|\Theta_i - \Theta_j| <= Threshold\_of\_\Theta$

      if $|\rho_i - \rho_j| <= Threshold\_of\_\rho$

$$\left|\begin{array}{lcl} \Theta_j & = & \dfrac{n_i\Theta_i + n_j\Theta_j}{n_i + n_j} \\ \rho_j & = & \dfrac{n_i\rho_i + n_j\rho_j}{n_i + n_j} \\ n_j & = & n_i + n_j \\ merging & = & 1 \end{array}\right.$$

      else

        $j = j + 1$

    if $merging = 0$

$$\left|\begin{array}{lcl} NbLines & = & NbLines + 1 \\ \Theta_{NbLines} & = & \Theta_i \\ \rho_{NbLines} & = & \rho_i \\ n_{NbLines} & = & n_i \end{array}\right.$$

$$\left|\begin{array}{lcl} merging & = & 0 \\ j & = & 1 \end{array}\right.$$

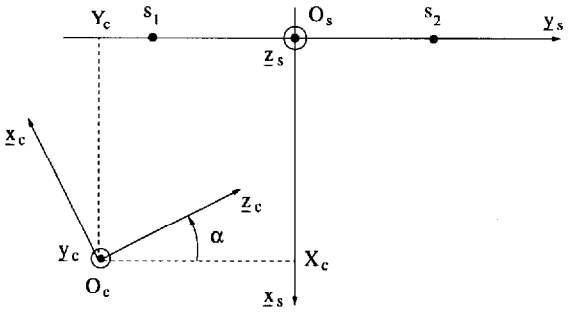

Figure 6: Mark and camera reference frames

6 Position Estimation

To determine the actual position of the mobile robot by using the vision system, in this paper it is considered the case in which some *ad hoc* land-mark images are placed in the environment, in location that are known *a priori* to the navigation system. The more complete and realistic case in which the robot position estimation and calibration is carried out by using the "true" environment alone, will be considered in the future. The case considered here can be of interested in a number of real applications, in which *ad hoc* reference images can be easily installed, e.g., in the case of mobile robots used in hospitals.

In particular, in this paper it is assumed that pairs of "star" marks (see Figure 8) are fixed at relevant, known, positions along the walls, and used to re-calibrate the robot navigation system. The centers of the two marks (of each pair), as projected on the image, will correspond to two points $S1$ and $S2$, that can be determined based on the algorithm outlined in the following. Then, by means of the positions S_1 and S_2 (in pixel coordinates), using the *a priori* knowledge of the (absolute) coordinates of the marks s_1 and s_2, the position of the camera reference frame R_c, hence the robot position, with respect to the environment can be determined.

Let $R_s = (O_s, \underline{x}_s, \underline{y}_s, \underline{z}_s)$ be a reference frame associated to a pair of marks, and placed such that the coordinates of the centers of the two "star" marks, with respect to R_s, are $s_1 = (0, -l, z_s)$ and $s_2 = (0, l, z_s)$, where z_s is the elevation of the centers of the two marks with respect to ground (See Figure 6). The position of the two mark centers s_1 and s_2, with respect to the camera reference frame R_c, is given by:

$$s_1 = \begin{bmatrix} X_c\cos\alpha + (Y_c + l)\sin\alpha \\ h \\ X_c\sin\alpha - (Y_c + l)\cos\alpha \end{bmatrix}$$
$$s_2 = \begin{bmatrix} X_c\cos\alpha + (Y_c - l)\sin\alpha \\ h \\ X_c\sin\alpha - (Y_c - l)\cos\alpha \end{bmatrix} \quad (5)$$

where h is the elevation of the marks with respect to the camera reference frame (it is assumed that all the vertical coordinates remain constant during robot operations),

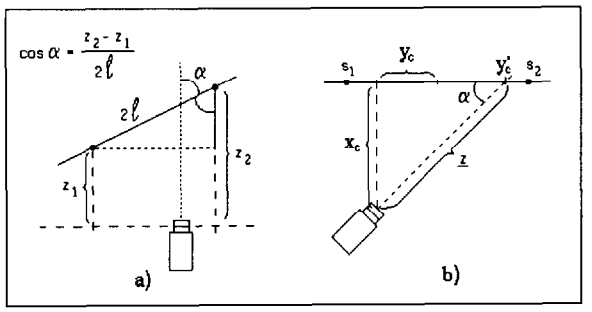

Figure 7: Position estimation

and α is the angle between the axes $\underline{z}_c$ and $\underline{y}_s$. Based on equations (5), and (2), and by means of the knowledge of the positions $S_1 = (X_1, Y_1, f)$ and $S_2 = (X_2, Y_2, f)$, the robot position can be easily determined. Two cases are considered: $Y_1 \approx Y_2$, and $Y_1 \neq Y_2$.

Case 1. $Y_1 \approx Y_2$

Let Y be the average value of Y_1 and Y_2. Then, the orientation angle α is close to $\pi/2$ ($\alpha \approx \pi/2$).

The origin O_c of the camera reference frame R_c has the following position with respect to the mark reference frame R_s:

$$X_c = \begin{cases} \dfrac{2lf_X}{X_2 - X_1} \\ \dfrac{f_Y}{Y}h \end{cases} \qquad Y_c = -l\frac{X_1 + X_2}{X_2 - X_1} \tag{6}$$

where x_c is determined by any one of the two equations, or by their average values.

Case 2. $Y_1 \neq Y_2$

The camera (hence the robot) is not along the axis x_s, and therefore $\alpha \neq \pi/2$.

The second equation in (2) allows to easily compute the z coordinates, z_1 and z_2, for the center of the two marks, and then, the distance $2l$ between the center of the two marks being known, it is easy to compute the orientation angle α (see Figure 7). Finally, in order to determine the camera position in the plane, the following simple procedure can be followed:

1. the distance $\underline{z}$ of the camera from the vertical plane comprising the two marks is computed as

$$\underline{z} = z_1 - \frac{X_1}{(X_2 - X_1)}(z_2 - z_1), \tag{7}$$

2. next, the coordinate of the origin O_c of the camera reference frame is computed as:

$$X_c = \underline{z}\sin\alpha \quad Y_c = Y_c' - \underline{z}\cos\alpha \tag{8}$$

where $Y_c' = (-l) - \frac{X_1}{(X_2 - X_1)}2l$.

Once the position and orientation of the camera reference frame has been determined, the position and orientation of the robot can be computed in an easy manner.

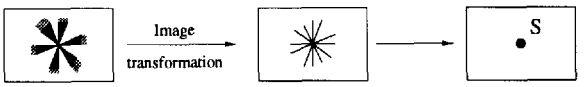

Figure 8: Image transformation for a mark

A "star" mark comprises 6 *black* areas ($c(X,Y) = 255$), the *white* background ($c(X,Y) = 0$) and 6 *grey* areas ($c(X,Y) < 200$) (see Figure 8). The grey areas allow to eliminate the 6 lines corresponding to the borders between the black areas and the background, in view of the threshold mechanism (the threshold has been chosen slightly higher than 200 in the experiments). Once all the 6 lines comprising a mark have been determined, the (average) point, intersection of all the lines, is computed (see Figure 8). To simplify the computations, the two marks are placed with the centers aligned along a common horizontal line.

7 Room-to-room path traveling

In order to allow accurate (and successful) navigation of a mobile robot along a room-to-room path segment, a precise knowledge of robot position, at the beginning of the segment, is required. Hence, the navigation system presented in this paper is such that just before the execution of the door "crossing" task, it performs a position verification and correction procedure. For this purpose, two marks are placed on one side of the door, 60 cm (i.e., $l = 30$ cm) away one from the other, and close to the door. Both marks are placed on the same door side, so that, for most common doors, the inter-mark distance l is not too large, hence making it possible for the robot to stop, for the position correction procedure, quite close to the marks. As a matter of fact, for a typical value for the parameter f_X (e.g., $f_X = 650$), and with a door 1m width, the camera should be at least 1.6 meters away.

The procedure to travel along a room-to-room path segment can be summarized as follows.

Step 1. The robot, based on previously planned path, and by only using odometry data, stops in front of the marks, at a planned distance (camera-marks) of about 1 meter (the actual position may be also quite different, depending on accumulated errors).

Step 2. The navigation system, and in particular the vision sub-system, determines the actual robot position, by means of a snapshot of the scene, carrying on the two mark image.

Step 3. Once the position has been corrected, the door status is determined by using the US captors.

Step 4. If the door is open, the robot goes through the door, otherwise a proper warning routine is executed, e.g., to require operator assistance.

Measured position			Estimated position		
x_c	y_c	α	$\hat{x}_c$	$\hat{y}_c$	$\hat{\alpha}$
103	2	90°	104	1	90°
95	15	90°	95	13	90°
133	−10	90°	135	−11	90°
78	6	90°	80	5	90°
101	−7	90°	99	−7	90°
80	−3	90°	81	−1	90°
158	−19	90°	156	18	90°

Table 1: Position estimation error in the case $Y_1 \equiv Y_2$

Measured position			Estimated position		
x_c	y_c	α	$\hat{x}_c$	$\hat{y}_c$	$\hat{\alpha}$
113	98	130°	109	96	131°
90	26	108°	92	27	109°
98	55	119°	98	51	118°
91	86	125°	90	84	125°
126	115	129°	130	113	127°
92	11	101°	95	13	100°
107	85	131°	111	88	133°
88	81	131°	89	81	131°

Table 2: Position estimation error in the case $Y_1 \neq Y_2$

8 Experimental results

The procedure for estimating robot position and orientation, based on equations (6) and (8), has been implemented and tested on the mobile platform LABMATE, available at the University of Rome "Tor Vergata". The tests have been carried out as follow.

Step 1. The robot has been placed close to the reference marks, in a position such that the marks lie in the camera image.

Step 2. An image is captured and processed to locate the two marks, and the two points S1 and S2 are computed.

Step 3. Finally, equations (6) and/or (8) are used to calculate the robot position.

The results for the cases corresponding to robot positions for which $Y_1 = Y_2$ are reported in Table 1, whereas the results for the cases $Y_1 \neq Y_2$ are shown in Table 2 (In both tables, the x and y coordinates are in meters, the angle are in degrees). In both cases, the position estimation error achieved by means of the vision system is smaller than 5 cm. Since the error in position estimation by means the odometry (encoders) is in the order of 1%, the robot position could be corrected by means the vision system about every 10 m (i.e., when an error in the order of 10 cm has been accumulated by means of the odometry system).

Acknowledgments

This work was partly supported by ASI funds. The work of the first author, Dr. Bétourné, has been supported by European Community, in the framework of the ERNET program (CHRX-CT93-0381).

References

[1] H. Durrant-White, *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Norwell, MA: Kluwer Academic Publishers, 1988.

[2] I. Cox and G. Wilfong, *Autonomous Robot Vehicles.* New-York, NY: Springer-Verlag, 1990.

[3] J. Latombe, *Robot Motion Planning.* Boston, MA: Kluwer Academic Publisher, 1990.

[4] M. Kabuka and A. Arenas, "Position verification of a mobile robot using standard pattern," *IEEE J. of Robotics and Automation*, vol. 11, pp. 505–516, 1987.

[5] D. Green, J. Sasiadek, and G. Vukovich, "Path tracking, obstacle avoidance and position estimation by an autonomous, wheeled planetary rover," in *IEEE Int. Conference on Robotics and Automation*, (San Diego, CA), pp. 1300–1305, 1994.

[6] R. Talluri and J. Aggarwal, "Mobile robot self-location using model-image feature corrrespondence," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 63–77, 1996.

[7] A. Kosaka and C. Kak, "Fast vision-guided mobiel robot navigation using model-based reasoning and prediction of uncertainties," *CVGIP: Image Understanding*, vol. 56, no. 3, pp. 271–329, 1992.

[8] B. Espiau, F. Chaumette, and P. Rives, "Une nouvelle approche de la relation vision-commande en robotique," tech. rep., Rapport de Recherche INRIA, Mars 1990.

[9] D. Ballard, "Generating the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, 1988.

[10] J. Princen, J. Illingworth, and J. Kittler, "A formal definition of the Hough transformation: Properties and relationship," *J. Math. Imaging Vision*, vol. 1, pp. 153–168, 1992.

[11] A. Bétourné, S. Nicosia, and P. Valigi, "A vision based obstacle detection system for mobile robots," in *Advances in Robotics. The ERNET Perspective. Proc. of the Research Workshop of ERNET, European Robotics Network, Darmstadt, Germany, 9-10 Septermber 1996* (C. Bonivento, C. Melchiorri, and H. Tolle, eds.), pp. 173–182, Singapore: World Scientific Publishing Co. Ltd., 1996.

Measured position			Estimated position		
x_c	y_c	α	$\hat{x}_c$	$\hat{y}_c$	$\hat{\alpha}$
103	2	90°	104	1	90°
95	15	90°	95	13	90°
133	−10	90°	135	−11	90°
78	6	90°	80	5	90°
101	−7	90°	99	−7	90°
80	−3	90°	81	−1	90°
158	−19	90°	156	18	90°

Table 1: Position estimation error in the case $Y_1 \equiv Y_2$

Measured position			Estimated position		
x_c	y_c	α	$\hat{x}_c$	$\hat{y}_c$	$\hat{\alpha}$
113	98	130°	109	96	131°
90	26	108°	92	27	109°
98	55	119°	98	51	118°
91	86	125°	90	84	125°
126	115	129°	130	113	127°
92	11	101°	95	13	100°
107	85	131°	111	88	133°
88	81	131°	89	81	131°

Table 2: Position estimation error in the case $Y_1 \neq Y_2$

8 Experimental results

The procedure for estimating robot position and orientation, based on equations (6) and (8), has been implemented and tested on the mobile platform LABMATE, available at the University of Rome "Tor Vergata". The tests have been carried out as follow.

Step 1. The robot has been placed close to the reference marks, in a position such that the marks lie in the camera image.

Step 2. An image is captured and processed to locate the two marks, and the two points S1 and S2 are computed.

Step 3. Finally, equations (6) and/or (8) are used to calculate the robot position.

The results for the cases corresponding to robot positions for which $Y_1 = Y_2$ are reported in Table 1, whereas the results for the cases $Y_1 \neq Y_2$ are shown in Table 2 (In both tables, the x and y coordinates are in meters, the angle are in degrees). In both cases, the position estimation error achieved by means of the vision system is smaller than 5 cm. Since the error in position estimation by means the odometry (encoders) is in the order of 1%, the robot position could be corrected by means the vision system about every 10 m (i.e., when an error in the order of 10 cm has been accumulated by means of the odometry system).

Acknowledgments

This work was partly supported by ASI funds. The work of the first author, Dr. Bétourné, has been supported by European Community, in the framework of the ERNET program (CHRX-CT93-0381).

References

[1] H. Durrant-White, *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Norwell, MA: Kluwer Academic Publishers, 1988.

[2] I. Cox and G. Wilfong, *Autonomous Robot Vehicles.* New-York, NY: Springer-Verlag, 1990.

[3] J. Latombe, *Robot Motion Planning.* Boston, MA: Kluwer Academic Publisher, 1990.

[4] M. Kabuka and A. Arenas, "Position verification of a mobile robot using standard pattern," *IEEE J. of Robotics and Automation*, vol. 11, pp. 505–516, 1987.

[5] D. Green, J. Sasiadek, and G. Vukovich, "Path tracking, obstacle avoidance and position estimation by an autonomous, wheeled planetary rover," in *IEEE Int. Conference on Robotics and Automation*, (San Diego, CA), pp. 1300–1305, 1994.

[6] R. Talluri and J. Aggarwal, "Mobile robot self-location using model-image feature corrrespondence," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 63–77, 1996.

[7] A. Kosaka and C. Kak, "Fast vision-guided mobiel robot navigation using model-based reasoning and prediction of uncertainties," *CVGIP: Image Understanding*, vol. 56, no. 3, pp. 271–329, 1992.

[8] B. Espiau, F. Chaumette, and P. Rives, "Une nouvelle approche de la relation vision-commande en robotique," tech. rep., Rapport de Recherche INRIA, Mars 1990.

[9] D. Ballard, "Generating the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, 1988.

[10] J. Princen, J. Illingworth, and J. Kittler, "A formal definition of the Hough transformation: Properties and relationship," *J. Math. Imaging Vision*, vol. 1, pp. 153–168, 1992.

[11] A. Bétourné, S. Nicosia, and P. Valigi, "A vision based obstacle detection system for mobile robots," in *Advances in Robotics. The ERNET Perspective. Proc. of the Research Workshop of ERNET, European Robotics Network, Darmstadt, Germany, 9-10 Septermber 1996* (C. Bonivento, C. Melchiorri, and H. Tolle, eds.), pp. 173–182, Singapore: World Scientific Publishing Co. Ltd., 1996.