

## DOOR CROSSING AND STATE IDENTIFICATION USING ROBOTIC VISION

**Eliana P. L. Aude\*, Ernesto P. Lopes\*\*, Cristiano S. Aguiar\*, Mario F. Martins\***

*\*Computer Center  
Federal University of Rio de Janeiro  
P.O. Box 2324, R.J. 20001- 970  
Rio de Janeiro, BRAZIL  
elaude@nce.ufrj.br*

*\*\*Institute of Mathematics  
Federal University of Rio de Janeiro  
P.O. Box 2324, Rio de Janeiro, R.J. 20001- 970  
BRAZIL  
lopes@cos.ufrj.br*

**Abstract:** This paper presents a new algorithm used to identify door states autonomously and without any previous information about door aperture, color and texture or about the location of the robot. Our approach imposes no constraint on the aperture of the door and enables a robot to identify doors in any state ranging from totally open, widely open and slightly open to closed using visual information. In this proposed solution, the robot is also able to judge whether the aperture of the door is suitable for crossing and to drive itself safely across the door.

*Copyright © 2006 IFAC*

**Keywords:** Robotics Navigation, Door Identification, Door-crossing, Path Planning and Computer Vision

### 1. INTRODUCTION

The greatest challenge in robotics today is to create robots which are able to perform increasingly complex tasks autonomously and with little previous knowledge about the environment around them. With this in mind, we formulated an approach to the door crossing problem which gives the robot a minimum of previous information and imposes no constraint on the aperture, color, texture and location of the door in relationship to the robot.

A series of approaches have been proposed by different research groups in the field of computer vision to the problem of recognizing doors by visual characteristics such as color, form and texture.

The works referred in (Amir, 1999), (Tomono, 2000) and (Martinez, *et al.*, 2004) have proposed approaches to door recognition based on color where objects with a certain color pattern are isolated from

the rest of the scene. However, only closed doors are recognized in these works.

(Stoeter, *et al.*, 2000) proposes a navigation system to be embedded in a robot which has to avoid obstacles and reach the doors to perform its tasks. It searches for doors by analyzing images features and uses sonars to check whether it is open.

(Monasterio, *et al.*, 2001) also uses sonar combined with visual information to detect doors and cross it. They design a controller based on Neural Network to make the robot cross the door. The solution identifies two door states: open door and closed door.

In (Libuda and Kraiss 2000), a stereo vision system is used to identify natural landmarks to control wheelchair.

In (Thurn, 1998), (Lazkano, *et al.*, 2003) and (Xiang and Santos, 2004) methods to assist robots in crossing doors are discussed. These works are based on the use of sonar and Infra Red sensors to detect

totally open or closed doors, but they do not consider partially open doors and provide no assessment of whether a door's aperture is suitable for crossing. In our environment doors are dynamic objects which can be found in any state of aperture from closed, slightly open, widely open to totally open. Therefore, approaches which do not consider these possibilities can't be applied to our solution. Furthermore, our doors present widely varying patterns of color and texture, which makes the use of color based recognition approaches ineffective.

This paper proposes a new algorithm to enable a robot to autonomously find and cross doors within an unknown environment. The robot performs the following tasks to cross doors:

- Detects the presence of doors;
- Recognizes the door's state;
- If the door's state is classified as "slightly open", decides if the door opens to the left or to the right, if the door's aperture length is suitable for crossing and, finally;
- Drives itself across the door after sending the trajectory to be controlled to the multivariable control module.

It must be stressed that the robot's only knowledge about the environment is the door's width.

The robot, shown in Fig.1, searches for doors by looking at the floor with a tilt of 12 degrees in relation to the horizontal. Its onboard camera is 65 cm high. In order to detect the presence of doors and their states the algorithm firstly, using the Hough Transform, looks for two or more long vertical lines which take up more than three quarters of the image height, are separated by more than 20 pixels and are at an angle of  $90^\circ \pm 5^\circ$  in relation to the base of the image. When it views a door, the robot is located near it. The equation of the baseboard is obtained through the application of a sliding correlation window beside the delimiting walls or the door's frame. A technique similar to the hotelling transform described on (Gonzalez and Woods, 1993) is employed. By analyzing the baseboard line's slope, its intersection with the vertical lines, and the result of the application of an ordered sequence of correlation windows, the robot is able to perform the tasks listed above in order to cross the door. This is further explained in section 3.

In section 2 of this paper, the approach to detecting the door's frame and the proposed method for finding the baseboard line are described. In section 3, the robot's recognition of door states and decision on how to cross the door are discussed. Section 4 presents experimental results illustrating the performance of the proposed algorithm. Finally, Section 5 presents the main conclusions of this paper.



Fig. 1. Robot looking for the door

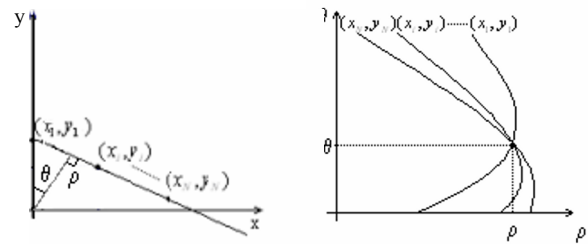
## 2. THE IMPORTANT FEATURES USED TO SOLVE THE ADDRESSED PROBLEM

Two main features are employed in order to detect doors and to establish their state of aperture: identification of long straight lines and determination of the baseboard's angle and position.

### 2.1 .The Adopted Approach to Detecting Door Frames

In order to detect door frames, the original image is firstly filtered by a Gaussian and Sobel Filter and then a Hough Transform is applied to it. The Hough Transform is a good technique for finding long straight lines which contain gaps. It maps each point  $(x, y)$  from the original data space onto a sinusoidal function in  $(\rho, \Theta)$  space or Hough space, where  $\rho$  gives the line's perpendicular distance from the origin and  $\Theta$  its slope. The points belonging to a straight line in the data space are transformed into sinusoidal waves in the Hough space. The intersection of the waves is a fixed point  $(\rho, \Theta)$ . This is shown in Fig. 2 and can be verified by equation 1, which defines the Hough Transform, by fixing  $(\rho, \Theta)$  and obtaining a straight line.

Fig.2. Right, the original data space. Left, the Hough



plane

$$\rho = x \cos(\theta) + y \sin(\theta) \quad , \quad (1)$$

where:  $-90^\circ \leq \theta \leq 90^\circ$  and  $0 \leq \rho \leq \sqrt{(m^2 + n^2)}$  ,

m, n are the image width and height

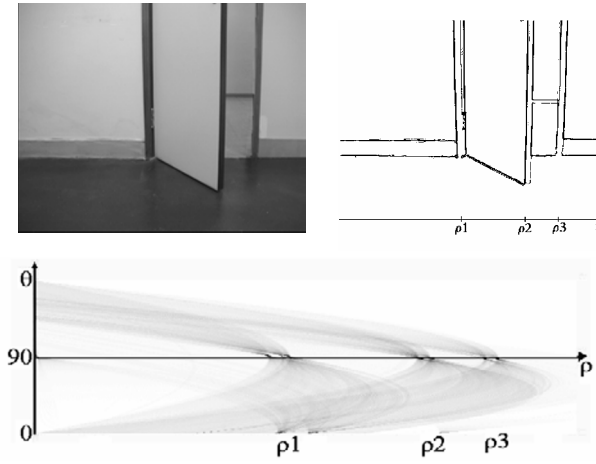


Fig.3. Captured Image (Top Left); Gradient image (Top Right) and finally, the Hough Transform Table

## 2.2 .Method for finding the baseboard line and the door's aperture.

The classification of the different door states depends upon the knowledge of the equation of baseboard straight line. This knowledge is obtained through the application of a sliding correlation window beside the delimiting walls or the door's frame. The sliding correlation window is also used, as will be discussed in section 3, when the robot realizes that the door is slightly open and needs to know if the door opens to the left or to the right.

The cross-correlation coefficient calculated for the contour points within the window is given by equation 2.

$$r = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)^2 \frac{1}{n-1} \sum_{i=1}^n (y_i - \mu_y)^2}} \quad (2)$$

where:

$x_i, y_i$  are x and y coordinates of contour points present within the sliding window,  
 $\mu_x, \mu_y$  are the mean values of the contour points coordinates present within the sliding window,  
n is the number of contour points within the sliding window.

$$y = \pm r \left( \frac{\sigma_y}{\sigma_x} \right) x + \mu_y \mp r \left( \frac{\sigma_y}{\sigma_x} \right) \mu_x \quad \text{or} \quad y = (tg\theta)x + h \quad (3)$$

where:

$$r = \pm 1$$

$\sigma_x, \sigma_y$  are the standard deviation of x and y coordinates

$\theta$  is the baseboard angle

The correlation mask is implemented as a square window with size 40x40 pixels as shown in Fig.4. Once the robot identifies two or more vertical lines (meaning that a door is present in the scene), the window scans the image from bottom to top in the region before the first and after the last long vertical line.

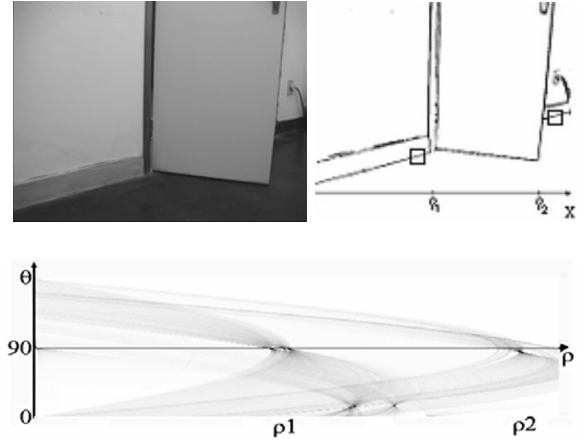


Fig. 4. The parameters which identify the baseboard line are found by scanning the image with a correlation mask shown at the top left corner .

The baseboard is located where one of the following situations arises: 1) the window reads correlation coefficients close to 1 in both windows or 2) The correlation coefficients are zero, the standard deviation in x is high and the standard deviation in y is low. In case 2, the baseboard line is horizontal. Once the baseboard is found, its inclination  $\theta$  is determined using equation 3. Table 1 presents the baseboard parameters calculated for the image shown in Fig.4 using equations 2 and 3.

Table 1. Parameters obtained from the left and right sliding windows used to calculate the baseboard  $\theta$  slope

Left window	Right window
$\mu_{1x}$ 235,40	$\mu_{1x}$ 591,66
$\mu_{1y}$ 391,73	$\mu_{1y}$ 286,70
$\sigma_{1x}^2$ 33,40	$\sigma_{1x}^2$ 36,36
$\sigma_{1y}^2$ 3,06	$\sigma_{1y}^2$ 3,09
r1 -0,930	r1 -0,874
$\theta_{1^\circ}$ -15,72	$\theta_{1^\circ}$ -14,31

By looking at the slope of the baseboard, as shown in Fig.5 the robot can determine if it is bent to the left, to the right or is located at 90 ° in relation to the baseboard line.

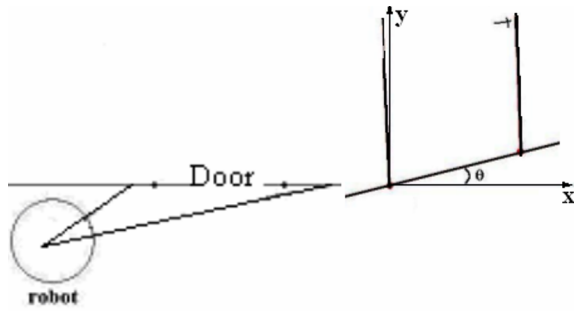


Fig.5. Robot looks to the door bent to the left. The captured image shows a baseboard with positive  $\theta$  angle.

### 3. THE REASONING FOLLOWED BY THE ROBOT TO IDENTIFY DOOR STATES AND THE APPROACH TO CROSSING DOORS.

#### 3.1 .Determination of Door State.and if the robot is inside or outside of the room

As stated before, the robot firstly searches for doors by analyzing each image taken by its onboard camera. This analysis consists of applying Hough Transforms to the image and determining if there are two or more vertical lines in it which occupy more than three quarters of the image height, are separated each other from by more than 20 pixels and are at an angle of  $90^\circ \pm 5^\circ$  in relation to the base of the image. As soon these image features are identified, the robot recognizes that it is viewing a door which is located at a near distance. Next, the equation of the baseboard is obtained through the application of a sliding correlation window which is applied beside the long straight line located furthest to the left of the image and beside the one located furthest to the right of the image.

To correctly determine the door state, the following situations must be considered: the robot can be facing the door from inside the room or from the corridor, these two situations are shown in figure 6.If it is inside the room, it may view a door located close to a corner or away from it.

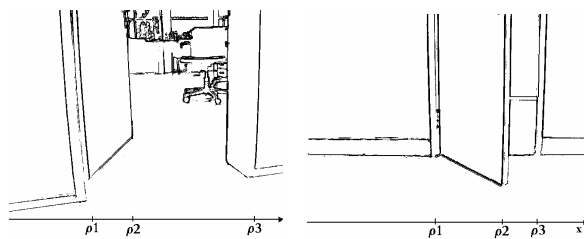


Fig.6 Left: Robot outside the room the door's base appears at the image above the baseboard. Right: Robot inside the room door's base appears below the baseboard

These three different situations are illustrated in Fig. 7a, 7b, 7c, which depict the image contour of a closed door viewed by the robot when it is inside the room and the door is away from a corner (Fig. 7a.), when the robot is facing the door from the corridor (Fig. 7b.) and when the robot views a door close to a corner from inside the room (Fig. 7c.). Fig. 7d, 7e,

7f, describe similar situations for slightly open doors and Fig. 7g, 7h, 7i, model these situations for widely open and completely open doors.

Firstly, the sliding correlation window is applied beside the long straight line located furthest to the left of the image and beside the one located furthest to the right of the image contour, as shown in Fig. 7. The window is applied from the bottom to the top of the image, until a non-vertical line is found. If both windows find a non-vertical line at the same angle, this line is identified as the baseboard. If the lines found by the two windows are at different angles, the robot recognizes that it is viewing a door close to a corner (Fig. 7c, 7f, 7i).

A second pair of correlation windows is then applied beside the long vertical lines immediately next to the ones to which the first pair of correlation windows was applied. If either of the second correlation windows finds a non vertical line which has the same slope as the baseboard line or which is at an angle to the baseboard which is positive in the clockwise direction, the robot recognizes that it is facing a door from inside a room (Fig. 7a, 7c, 7d, 7f, 7g, 7i, 7j). If either of the windows finds a non vertical line at an angle which is positive in the anticlockwise direction, the robot realizes that it is viewing a door from the corridor (Fig. 7b, 7e, 7h).

A third pair of correlation windows is applied if the robot is viewing the door from the corridor or if it is viewing a door close to a corner and if only one of the correlation windows of the second pair found a non vertical line with the same slope as the baseboard (Fig. 7f, 7i).

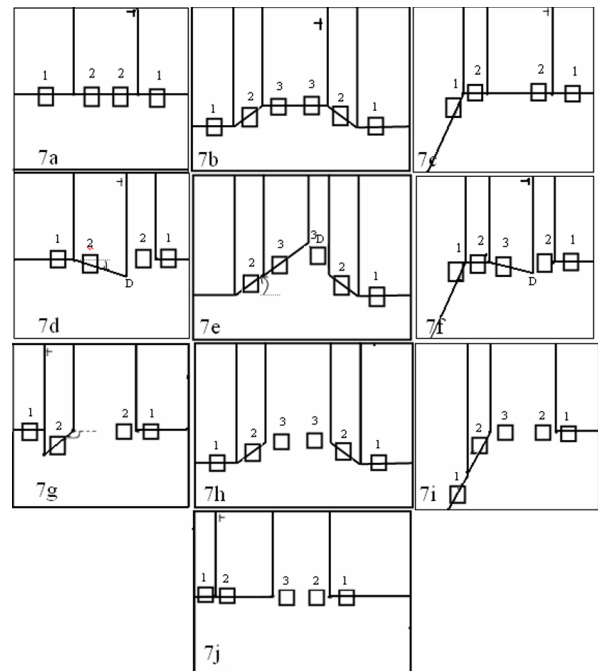


Fig. 7. Possible door states and locations.

After applying two or more pairs of correlation windows, the robot can determine the state of the door by answering the following questions:

- A – Does the end point of both the first and last long vertical lines belong to the baseboard line?
- B - Is the angle determined by the second pair of correlation windows 0 or positive in the clockwise or anticlockwise direction in relation to the baseboard line?
- C – Does the third pair of correlation windows applied identify a non vertical line? If so, at what angle is this line?

The door is determined to be closed if: a) Condition A is true and condition B indicates a line with the same slope as the baseboard (Fig. 7a). b) Condition A is true and condition B indicates at least one line with a slope angle that is positive in the anticlockwise direction and greater than that of the baseboard and condition C indicates a line with the same slope as the baseboard (Fig. 7b). c) Condition A is true and condition B indicates a line with the same slope as the baseboard (Fig. 7c).

Similarly, it is concluded that the door is slightly open if one of the following is true: a) Condition A is true and condition B indicates at least one line with a slope angle that is positive in the clockwise direction and greater than that of the baseboard (Fig. 7d). b) Condition A is true, condition B indicates a line with a different slope from that of the baseboard and at an angle which is positive in the anticlockwise direction and condition C indicates a line with slope different from that of the baseboard at an angle which is positive in the anticlockwise direction (Fig. 7e). c) Condition A is true, condition B indicates a line with the same slope as the baseboard line and condition C indicates a line at an angle which is positive in the clockwise direction and which has a different slope to that of the baseboard line. (Fig 7f).

A similar reasoning is applied to recognize widely open (Fig. 7g, 7h, 7i) and completely open doors (Fig. 7j). When the robot is not facing the door head on, the same reasoning is followed now considering the inclination of the baseboard.

### 3.2 .Assessment of Door Aperture and Approach to door Crossing

Once the robot is at  $90^\circ$  to the baseboard line, it can easily calculate the aperture length ( $d_a$ ) and angle of aperture ( $\alpha$ ) of the door by using the known width of the door and measuring distance  $x$  from the image (shown in Figure 8).

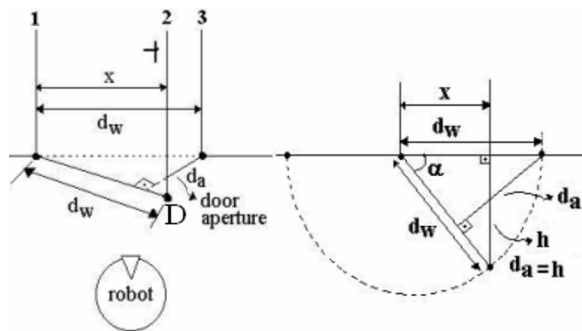


Fig. 8. Door aperture length and aperture angle calculation

If the aperture is suitable for crossing, the robot will cross the door. The robot will firstly determine whether it is to the left or to the right of point D (Fig. 8). If it is to the left of point D, it will rotate anticlockwise through an angle of  $(90 - \alpha)^\circ$  and translate to the right at  $90^\circ$  to its principal axis until it can only see two vertical lines of the door frame (i.e. it is aligned with the door (Fig. 10)). It will then

translate a further distance of  $\frac{d_a}{2}$  in the same

direction. The robot will now be at the right position and angle to cross the door and will drive itself forward to do so until it reaches the minimum distance  $d_a$  where it rotates in relation to its rear wheel clockwise through an angle of  $(90 - \alpha)^\circ$ .

When the robot is outside the room and the door state is slightly open it has to perform the reverse path. Note that if the robot is to the right of point D, it will follow the same sequence of movements but will translate to the left.

## 4. EXPERIMENTAL RESULTS

The following real-time experiment shows the robot executing of our algorithm.

The robot reaches the door at angle to the baseboard (Fig.9) and calculates the baseboard line equation. Then, the door state is determined as slightly open. The Table 2 shows the coefficients of the lines identified by the three correlation windows shown in Fig. 9.

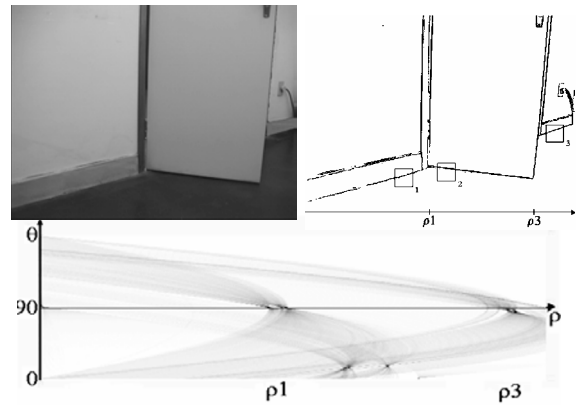


Fig. 9 The top right figure shows the captured image. Below, the Hough Transform shows two long vertical lines. In the top left figure, the first pair of correlation windows evaluates the baseboard inclination and the second window indicates that the door is slightly open.

Table 2: Correlation Windows Coefficients

Left Window (1)		Middle Window (2)		Right Window (3)	
$\mu_{1x}$	259,80	$\mu_{1x}$	340,88	$\mu_{1x}$	610,12
$\mu_{1y}$	384,73	$\mu_{1y}$	371,37	$\mu_{1y}$	279,58
$\sigma_{1x}$	8,30	$\sigma_{1x}$	11,20	$\sigma_{1x}$	5,65
$\sigma_{1y}$	2,60	$\sigma_{1y}$	1,79	$\sigma_{1y}$	1,84
$r_1$	-0,93	$r_1$	0,78	$r_1$	-0,91
$\theta_1$	-16,17	$\theta_1$	6,84	$\theta_1$	-16,69

The robot will now follow a sequence of movements of translations and camera rotations (described in Section 3.2) to place itself at 90° to the baseboard line, as shown in Figure 8. It will then find the length of the aperture of the door and the aperture angle and decide whether it is physically possible to cross the door. Fig. 10 shows the values of the parameters to be measured from the image when the robot is located perpendicular to the baseboard. Knowing that the door's width is 80 cm and counting the number of pixels within the image corresponding to the horizontal distance between the door's external and internal vertical lines  $d_w$ , the angle  $\alpha$  and the minimum door aperture  $d_a$  are calculated. In this experiment these values are given by:

$$x = \frac{139}{215} * 80\text{cm} = 51.72\text{cm} \quad \text{and}$$

$$\alpha = \sin^{-1}\left(\frac{h}{D_w}\right) = \sin^{-1}\left(\frac{88}{215}\right) = 24.16^\circ$$

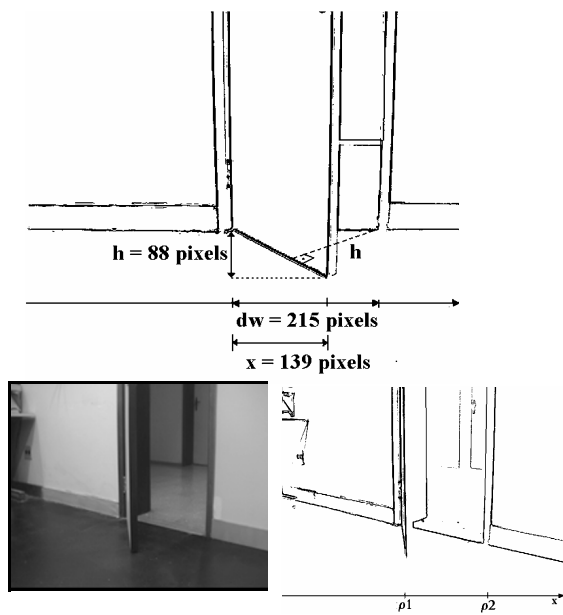


Fig.10. Top: the robot is at 90 degrees to the baseboard line and the image measurements are made. Bottom: The robot is aligned with the door and ready to cross it.

The robot's diameter is 50cm. The robot is firstly at an angle to the baseboard and so has to perform a series of translations to the right at 90° to its main axis and anticlockwise camera rotations until the camera reveals that the baseboard angle is 0 and so the robot is placed facing the door. The robot now calculates the door aperture  $d_a$  and the aperture angle  $\alpha$ . It will then rotate anticlockwise through an angle of  $(90 - \alpha)$  and its camera will rotate clockwise through an angle of  $\alpha$ .

The robot now translates until it is aligned with the door (Fig.10) and then translates to the right through a distance of  $\frac{d_a}{2}$  ( $d_a$  being the door's aperture length), Finally, the robot drives itself through the door.

## 5. CONCLUSION

A new algorithm for enabling a robot to autonomously recognize and cross doors without any previous knowledge of the environment was presented. The robot successfully identifies any door states, distinguishing between totally open, widely open, slightly open and closed doors. The solution also enables the robot to judge if the door aperture is suitable for crossing and to drive itself across the door. Real time experiments have shown promising results which demonstrate the successful application of the algorithm.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank FINEP and CNPq, Brazil, for supporting the development of this work.

## 7. REFERENCES

- Amir, Maynard, P. (1999): "Door Identification". <http://wwwformal.stanford.edu/eyal/cs223b/repo/rt.html>.
- Gonzalez, R.C, Woods, R.E, 1993 "Digital Image Processing", Addison-Wesley.
- Lazkano, E., Sierra, B., Rañó, I., Zarauz, I. (2003), "Door crossing behavior for a mobile robot using Bayesian Networks", *Proceedings of CIMCA*.
- Libuda, L., Kraiss, K. (2000). "Identification of Natural Landmarks for Vision Based Navigation", *Proceedings of the 2000 IEEE Int. Conf. on Robotics & Automation*, San Francisco, CA.
- Martínez, J. C., Sánchez, D., Suárez, B. P. (2004) "A Fuzzy Colour Image Segmentation Applied to Robot Vision". In *Conf. on Mechatronics & Robotics*.
- Monasterio, I., Lazkano, E., Rañó, I., Sierra, B. (2001). "Learning to traverse doors using visual information", *Systems Science Journal*, Vol. 27, 97-108.
- Stoeter, S.A., Mauff, L.F Papanikolopoulos, N.P. (2000). "Real-time Door Detection In Cluttered Environments". *Proceedings of IEEE Int. Symp. On Int. Control*, Greece, pp187-192.
- Thrun, S. (1998) "Finding Landmarks for Mobile Robot Navigation", *Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA)*.
- Tomono, M., Yuta, S. (2000). "Mobile Robot Navigation in Indoor Environments using Object and Character Recognition", *Proc. of the 2000 IEEE International Conf. on Robotics & Automation*, San Francisco, CA.
- Xiang, Z., Santos, V. M. F. (2004). "Locating and crossing doors and narrow passages for a mobile robot", *Proceedings of ICINCO*.