Learning to traverse doors using visual information

Iñaki Monasterio, Elena Lazkano*, Iñaki Rañó, Basilo Sierra

Department of Computer Science and Artificial Intelligence, University of the Basque Country, P.O. Box 649, E-20080 San Sebastián, Spain

Abstract

Mobile robots need to navigate in their environment in order to perform useful tasks. Doors appear in almost every office-like indoor environment and they have to be crossed often during the navigation process. We present in this paper a new approach that uses visual information to anticipate that a door has to be crossed. Combining then visual information with ultrasonic sensors, the robot approaches the door until an adequate distance is reached. Door traversing is then performed using sonar sensors. This paper describes the control architecture and the behaviors that have been implemented to obtain the door traversing behavior. Results and performance issues are explained. The experiments have been carried out with a B21 mobile robot.

Keywords: Door traversing behavior; Mobile robots; Neural networks; Visual door detection

1. Introduction

A mobile robot must be able to interact effectively with the environment. To safely navigate while performing a task it must be able to identify potentially dangerous situations not to damage itself or the rest of the environment (including persons). Doors present a serious obstacle for a B21 robot; often, the space left between the door side panels and the robot is so small that just a little rotation can make the robot collide.

Doors can be considered like narrow and short corridors during navigation and try to cross them balancing the open space found at both sides of the robot using sonar sensors. But some problems arise: if the robot is wandering around, it just decides to go to somewhere else where the free space is more significant; if the robot needs to follow some given orientation, i.e. if it is mandatory to go in the direction the door is, the door blade often misleads the sonar readings. But if the door has been identified, the robot can anticipate that it has to go through a very narrow passage and tackle it from a privileged central position. Although it is not a trivial task, it is easier to cross a door if the robot is just in front of it. But, in order to position itself in front of the door, the robot must know what a door is and how to pass through the door.

* Corresponding author.
E-mail addresses: cppmoiri@si.ehu.es, ccploare@si.ehu.es (E. Lazkano).

0378-4754/02 $ – see front matter © 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.
PII: S0378-4754(02)00027-7
There are some approaches to door identification. In [1] the color of the door panels is used to identify closed doors and this information is used for self-localization. The doors are extracted from the rest of the environment using their characteristic color (see [2] for a revision on color treatment). Once door candidates are found, only those with a significant mass are considered and Hough transform is applied to locate them. Then, if two lines are close to vertical but sufficiently apart, the object is labeled as a door. In [3] a ranger robot is used to launch smaller robots in environmental conditions that can be dangerous for humans. This ranger needs to recognize its position in the environment in order to launch the scouts at the proper place. They use vision to identify doors in a corridor, and once doors are placed, the robot looks if it is open with the sonars, to launch the scout to the new room. But the robot is not crossing doors. Some other approaches mention optical flow to locate moving vertical lines while going towards the door. There are different techniques to compute optical flow (see [4] for a review), more or less sensitive to light conditions, but all of them need heavily textured environments in order to achieve a good performance. Several works can be found about the recognition of general objects in images [5–7].

The work presented here is an initial attempt to include a door traversing behavior in a mobile robot navigation control architecture. We divide the traversing door behavior into two steps: door identification and approximation, and door crossing. The robot we have used for the experiments is a B21 RWI model [8]. Among many others, it is provided of a CCD camera sensor mounted on its top plate, and a ring of 24 sonar sensors. It has two internal 120 MHz PC’s, used to control the different devices.

The paper proceeds in Section 2 explaining each of the phases needed to perform the door traversing behavior. Section 3 introduces the control architecture we have used, and Section 4 presents the results and conclusions. Last section is devoted to future work.

2. Door traversing modules

The door traversing behavior can be separated into two subtasks: door identification and approximation in order to position the robot in a privileged position, and the door crossing step, i.e. the task of crossing the door. These two phases are described along the next two subsections.

2.1. Door identification and approximation task

As we said before, the door traversing behavior can be more efficient if the robot is facing the door opening. In order to locate the robot in front of the door, or very close to it, it is mandatory to identify a door. A color camera can be a very powerful sensor, but the necessary information must be properly extracted from the image. First, what an opened door is in an image must be defined. Taking advantage of the textureless door panels, when the door is opened the door opening appears as a squared noisy rectangular segment in the image, after applying an edge detector to the grey-scaled image. Many edge detectors can be applied [9,10], but due to the fact that not every lines but only the vertical straight lines are needed, the edge detector selected is just a vertical Sobel filter. This filter has also the added advantage that it is not computationally very expensive. The resultant image is again filtered first using a dilation and afterwards with a very simple algorithm that enhances the columns merging those columns separated by very thin spaces. The robot decides that there is a door if there exist a column rich wider than a given value (in our experiments, this value is set to 35 pixels). Figs. 1 and 2 show an original image taken

1 The size of the images are 120 × 160.
by the robot and the result of applying the different image processing filters needed to match a possible door.

Once the door space is extracted from the original color image, it must be defined where the door is located with respect to the robot position. To find the left and right columns that define the door opening limits, we use the indexes of the columns. The task can be implemented just as a feedback loop where the reference signal is the central column of the image and the variable to control is the center of the identified door. Thereby, if \( d_s \) is the space before the door opening starts and \( d_e \) is the width left at the right of the door opening, the sign of the rotational velocity, calculated as in Eq. (1)

\[
\text{sign}(\omega) = -\text{sign}(d_s - d_e)
\]

(1)

is enough to make the appropriate movements to maintain the rectangle centered in the image.

The variability of the light conditions make the positions that define the door fluctuate too much. Thereby, we have added a low pass filter to stabilize the column position values. If \( pos_s \) represents the number of the column where the door starts and \( pos_e \) is the column where the door ends, then the low pass filter is defined as follows:

\[
p\hat{pos}_s(t) = \alpha pos_s(t) + (1 - \alpha)p\hat{pos}_s(t - 1)
\]

(2)

\[
p\hat{pos}_e(t) = \alpha pos_e(t) + (1 - \alpha)p\hat{pos}_e(t - 1)
\]

(3)
This smooths the non-desirable abrupt changes in the door delimiting positions, when lighting conditions are not adequate. The robot keeps balancing the hypothesized door until a preventive distance is reached. In order to eliminate the false positives, i.e. to eliminate the cases where very noisy images carry the robot to the false belief that there is a door, we use a Neural Network [11] trained with a backpropagation algorithm that takes as input the normalized values of the eight front sonar sensors of the robot. This eight sonars cover an angle of 120°. To collect the data we have placed the robot in front of the door at a distance of 1 m (±20 cm) and read the sonars changing the angle to the center of the door from −30° to +30°. At 1 m our camera misses one of the edges of the door and the vision door detection as we implemented it might be discarded. We decided that the approximation to the door from that distance at higher angles had few chances of being crossed by our robot. Moreover, most of these cases should be neglected by the vision procedure. Fig. 3 shows the range of positions from where the collected data were labeled as: (a) door confirmation; (b) and (c) door rejection.

For discarding no doors, we get data from sonars in front of different kind of walls and corners and from positions from which the robot is misplaced with respect to the door. This neural network has
been trained with 2700 training patterns and tested with a set of 1300 patterns giving an accuracy of 99.47%.

Fig. 4 shows the structure of the door identification module. The Neural Network only confirms or rejects the output given by the vision module and thereby, helps to reject false positives and also to avoid positions from where the door crossing behavior becomes more difficult. The network output is only considered when a distance of about 1 m is detected by one of the front sonars. The robot then will halt to decide if it has or not a door in front.

2.2. Door traverse

This module makes the robot cross the door from a preventive distance of about 1 m. From this position the robot learns the actions it has to perform in order to successfully reach the end of the door. To learn the actions, we use as input data the 14 frontal sonar readings. We again use a Neural Network trained with the backpropagation algorithm. The structure of the ANN is the following: 14 input neurons, one for each of the sonar sensor values, 20 hidden input layers and 3 outputs. The first output corresponds to left rotation, the middle output means no rotation and the last one will order the robot to rotate to the right. Thereby, again only the sign of the rotation action is learned, and the magnitude of the rotation is considered fixed.

To collect data in order to learn the actions to cross the door, we have used the following methodology: we mark different trajectories on the floor, in which the same actions should be performed and collect sensor data while pushing the robot along that path. Fig. 5 shows the labels assigned to the different viewpoints of the robot with respect to the door. From angle $\theta$ the action to be carried out is “turn right”, from position with angle $\alpha$ the robot must not rotate at all, and if the sonar readings indicate that the angle with respect to the door is similar to $\Omega$, then the robot will “turn left”.

To train the network we have used a set of 22,240 input patterns. Two-thirds of them have been used to train the net and the rest to measure the accuracy, obtaining an efficiency of 97.77%.

3. Control architecture

To complete the door traversing behavior, all the modules have been organized in a control architecture. Fig. 6 shows the different states and the conditions needed in order to change from one state to another. The definition of each state is presented in the following section.
3.1. **INIT**

The main function of this module is to initialize the variables and start the system. The robot starts moving searching for possible doors. Once a door is recognized by the door-identification module, the state changes to try approaching the robot to the door.

3.2. **Door approximation**

While the robot does not miss the door and it does not detect something with its eight front sonars in a range of 1 m, the robot moves towards the center of the two detected edges of the door. If the door is missed, i.e. the door-detection ANN gives a negative answer, the robot stops and returns to INIT state, starting looking for new door candidates.

3.3. **Door confirmation**

Once the robot has followed the path of the door image until something is detected, it halts. While trying to guess if it has a door in front, the door-crossing behavior is let to act as an obstacle avoidance module, but with no translational velocity. This important fact lets the robot modify its orientation with respect to the door and helps the door-confirmation ANN to get better chance to detect the door opening, in case it exists. If the door-confirmation module dismiss the situation as a false positive, the robot makes a turn and goes back to the INIT module.

3.4. **Door crossing**

In this module, the robot is 1 m ahead facing the door. Door-crossing ANN starts deciding turnings, avoiding door edges and making precise steps for crossing the door. The translational speed of the robot
is set to a fixed value, empirically selected. The magnitude of the rotational velocity is also fixed, only the sign changes. Actually, this behavior is active during a time interval large enough to cross the door, but some wandering mechanism should be added in the future.

4. Experimental results and conclusions

We have separated the experiments in three subsets. First, the door identification using the vision module has to be evaluated. Secondly, it is necessary to measure the performance of the door-confirmation process, and the same thing applies to the door-crossing behavior.

4.1. Evaluation of the vision module

We have tried the door-identification vision module in different lighting conditions. After a lot of experimentation, we conclude that the door-recognition vision module works well under stable light conditions; variations in brightness and reflections alter notably the results of the vision module. In spite of this, when light conditions are adequate the vision module helps gratefully to identify door candidates and to reject false ones.

4.2. Evaluation of the door-confirmation module

For proving the door-confirmation module, we placed our B21 robot in several positions about 95 cm away from the door. Different headings have also been taken into account from each position as shown in Fig. 7. V1 and V3 correspond to the robot heading towards each edge of the door, and V2 represents

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door confirmation module accuracy with different viewpoints and different angles with respect to the door</td>
</tr>
<tr>
<td>$P_1$ (%)</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>V1</td>
</tr>
<tr>
<td>V2</td>
</tr>
<tr>
<td>V3</td>
</tr>
</tbody>
</table>

Fig. 7: Orientations to test the door-confirmation module.
that the robot is heading to the center of the door. Table 1 shows the percentages of success obtained from each position. The meaning of each $P_i$ position is represented in Fig. 8.

It can be deduced that this module works well between the incidence angles we used for training (less than $30^\circ$ from door center and 1 m far). It also works fine for no-door recognition. Door and no-door cases were clearly differenced. The robot also recognizes doors well from shorter distances than those used for training.

4.3. Evaluation of the door-crossing module

Once the robot is near the door and starts trying to cross it, the approaching angle makes the difference. Our robot only has a 20 cm margin for crossing the door, and for short distances, sonar waves do not have the adequate precision, not valid readings are obtained at distances shorter to 30 cm. In spite of that, and of the narrow incidence security angle, the robot manages to turn one side and the other until it goes through the door with great precision and few turnings.

It must be said that this module has shown a good behavior also when acting as an obstacle avoider in narrow corridors. Table 2 shows the performance rates obtained at the different positions. Only $P_3$ and $P_5'$ positions fall through, but these positions should be avoided by the door identification and approximation module.

4.4. Overall behavior

In order to evaluate the whole system, some experiments with the complete architecture have been made. The overall behavior is robust, the robot goes through the complete state sequence needed to

<table>
<thead>
<tr>
<th>$P_1$ (%)</th>
<th>$P_2$ (%)</th>
<th>$P_3$ (%)</th>
<th>$P_4$ (%)</th>
<th>$P_5$ (%)</th>
<th>$P_6$ (%)</th>
<th>$P_7$ (%)</th>
<th>$P_8$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>V2</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>V3</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table 2**
Door crossing accuracy for the module with different viewpoints and different starting angles

Fig. 8. Set of positions from where the experiments are done.
satisfactorily traverse the door. Figs. 9 and 10 show a complete sequence of the robot traversing the door successfully.

Even though the experiments made for each module separately have shown some weaknesses, they seem to be complemented showing satisfactory overall results. From large distances, if only vision is used, chances of missing the door are high. In spite of that, with the whole system the robot manages to detect the door from near 2 m away even from different trajectories, so that it has time enough to react. This distance can be larger under advantageous conditions, like good brightness and contrast of the door versus the background.
5. Future work

This work is an initial attempt to build an efficient door-traversing behavior. Although the experimental work done has shown good performance, more experimentation should be done to refine the vision module and the control architecture. A door color/texture recognizer could help to make the door identification process more robust and less sensitive to light conditions. Some work has been done in this direction, but with no success yet.

The door-traversing behavior should be complemented with a wandering behavior in order to act properly when getting out the door limits. Also, the compass information could help if some initial information about the location of the door is given to the robot.

Further work should also include mechanisms to learn not only the sign but also the magnitude of the translational and rotational velocities. Thereafter, the robot speed could adapt to the circumstances, slowing when entering a narrow passage and accelerating when free space is available.

References