Robotica (2005) volume 23, pp. 689–699. © 2005 Cambridge University Press
doi:10.1017/S0263574704001390 Printed in the United Kingdom

A multi-agent system architecture for mobile robot navigation based on fuzzy and visual behaviour

Rafael Muñoz-Salinas*, Eugenio Aguirre, Miguel García-Silvente and Moisés Gómez

Department of Computer Science and Artificial Intelligence, E.T.S. de Ingeniería Informática, University of Granada, 18071 Granada (Spain)

(Received in Final Form: November 26, 2004)

SUMMARY

A multi-agent system based on behaviour for controlling the navigation task of a mobile robot in office-like environments is presented. The set of agents is structured into a three-layer hybrid architecture. A high level of abstraction plan is created using a topological map of the environment in the *Deliberative* layer. It is composed by the sequence of rooms and corridors to traverse and doors to cross in order to reach a desired room. The *Execution and Monitoring* layer translates the plan into a sequence of available skills in order to achieve the desired goal and monitors the execution of the plan. In the *Control* layer there is a set of agents that implements fuzzy and visual behaviours that run concurrently to guide the robot. Fuzzy behavior manages the vagueness and uncertainty of the range sensor information allowing to navigate safely in the environment. Visual behavior locates a required door to cross and fixate it, indicating the appropriate direction to reach it. Artificial landmarks are placed beside the doors to show its position. The system has been implemented in a Nomad 200 mobile robot and has been validated in numerous experiments in a real office-like environment.

KEYWORDS: Multi-agent system; Mobile robot; Navigation; Behaviour.

I. INTRODUCTION

In the mobile robot navigation area, topological approaches for the representation of the environment have the advantage of manipulating the information at a high level of abstraction.[1] On the other hand, geometric representations usually need more computational effort and react slower.[2] Furthermore, geometric models are usually more dependent on the localization system requiring a high knowledge about the position of the robot to take the necessary decisions in the navigation. The meaning associated to nodes and arcs in the topological representations is not always the same. Whereas in reference [3] the topological map is obtained from the division of regions in a cell map, in reference [4] nodes represent important places from the point of view of the sensory system and the arcs paths between these places associated with a concrete control strategy.

Relating to behavior based navigation,[5,6] there are approaches that use topological maps for navigating. In some cases[7] the nodes represent distinguished places associated to the behavior, while in other works[8] the arcs also indicate the behavior needed to navigate between nodes. In these cases the navigation between two points is seen as the execution of a plan composed of a set of nodes joined by arcs or behavior to execute, where the nodes are sub-goals to achieve the desired plan. The abilities of the robot, like *follow wall*, *follow corridor*, *cross door* or *avoid obstacle*, are essential to achieve the generated plan. Because the plan generated is not highly detailed, the robot has to trust in these abilities to reach each sub-goal established in the plan. In these models the role of the sensory system is very important, because every behavior establishes a relation between perception and action leaded by its own intention.

In this work a multi-agent system architecture structured in three layers is presented. It uses a concurrent activation of both visual behaviors and fuzzy behaviors to carry out the navigation task of a mobile robot in office-like environments. The higher layer receives orders from an external operator and generates a high level of abstraction plan consisting in the sequence of rooms and corridors to transverse and doors to cross to go to a desired place. This plan is decomposed in the middle layer in a sequence of skills that the robot is able to perform. These skills arise from the interaction of visual and fuzzy behaviors implemented in the lower layer of the architecture. Fuzzy behaviour takes as input the information provided by the range sensors. They use fuzzy rules to manipulate the imprecision and vagueness of the sensor data allowing a safe navigation to accomplish each part of the plan. Visual behaviour gathers the visual information from a single camera placed at the top of our robot and allow the system to know the location of the doors of the environment and indicate the straight direction to cross them. The camera is placed over a pan-tilt unit (PTU). It allows to move the camera independently from the movements of the robot giving to vision an active role in our system. In order to ease the door detection, artificial landmarks with numerical information have been placed beside the doors. We assume that all doors are opened, because our robot cannot open them.

* Corresponding author. E-mail: salinas@decsai.ugr.es

The use of artificial landmarks has been successfully used in the literature as a mean to provide extra information to a robot in order to aid the self-positioning process as well as for developing other tasks.[9,10] Several designs of landmarks have been proposed depending on the purpose they are used for[11,12] and even the best arrangement in order to get minimum-error position has been studied.[13] We propose a simple artificial landmark that is placed besides the doors to aid the robot to know where the doors of the environment are. The landmark is easy to distinguish from the environment and has digits inside to uniquely identify the door that represents. Fast algorithms have been required in order to achieve real-time visual processing. Neural networks have been trained for either detecting the landmark and for classifying the digits.

Following sections deal with the multi-agent system architecture. First, we briefly describe the hardware and then a general vision of the architecture of the system is given. Following, the agents of the system are explained in detail. First, *Planner* agent that plans according to the desired goal. Then Monitor agent, that transform the plan in a sequence of skills that the robot is able to do. Finally, both *Vision* agent and *Navigation* agent are explained. The former develops the visual processing and the latter performs the navigation. In the section of experimental results we show an example of the whole system running in a Nomad 200 mobile robot. Finally, we offer some conclusions.

Fig. 1. Robot appearance.

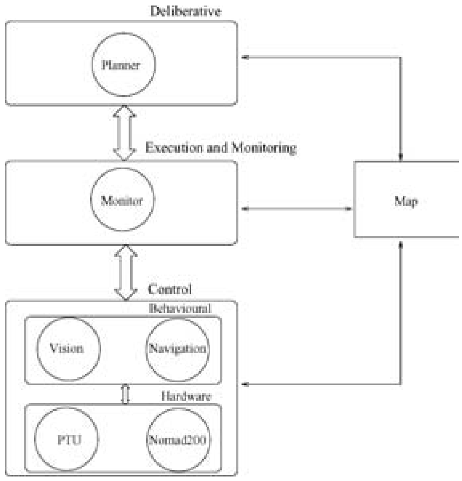

Fig. 2. Multi-agent system architecture.

II. MULTI-AGENT SYSTEM ARCHITECTURE

II.1. Hardware description

Our robot is a Nomad 200 that has been updated adding new devices. First of all, we have to increase the computational power to perform visual processing in real-time. Therefore we have added a mobile computer Pentium III running at 1.100 MHz that communicates with the robot via Ethernet at 10 Mbps using TCP/IP. There is a computer in the robot that is used only to run a daemon that receives data from the sensors and issues movement commands to the motors. The rest of the system runs in the mobile computer. To perform the visual processing, a pan-tilt unit (PTU) and a Sony EVI-401 video camera have been added. Both are connected to the mobile computer. The PTU can move 139 degrees to each side in the horizontal axis, while in the vertical axis it can move 47 degrees down-side and 31 degrees up-side. The camera is analogic, so it has been connected to a PCMCIA frame-grabber obtaining images at a frame rate of 25 fps. In Figure 1 there is shown the final appearance of the robot.

II.2. Multi-agent system overview

In this work a multi-agent system based on behaviours for controlling the navigation task of a mobile robot is presented. An agent is a software process aimed to get or keep a goal implicit in its own design. Our system has been designed as a set of agents organized in a three layer architecture with both reactive and deliberative capabilities in an hybrid architecture.[14] This methodology allows great robustness and an easy expansion of the system either adding new capabilities to the existing agents or adding new agents to the system. In Figure 2 the structure of the system is shown.

We have identified two different navigation tasks: The first one consists in navigating into a room to go to a certain place in it, and the second one consists in traveling from one room to another into the building. The first one has been developed in previous works[8,15,16] using fuzzy behaviours to move the robot following walls and a fuzzy perceptual model to build a topological map that contains some metric information of the environment. Using the perceptual model and the topological map the robot can navigate and determines the wall it is following and the corners of the room that are passed.

We want to focus this work on the ability to perform the second navigation task. When the robot must go from one room to another into a building a plan is created. It consists in a sequence of rooms and corridors to transverse and doors to cross to reach the desired room. This sequence is a high level of abstraction plan that turn into sub-goals for the system. For this purpose low precision about the position of the robot is required. The plan is not highly detailed but expressed at a high level of abstraction forcing the robot to trust in the abilities of the system to accomplish those sub-goals. The different agents of the system collaborate distributing

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

the whole navigation task establishing communication between them whenever it is necessary. Both visual and range sensor information are used in order to perceive the environment.

At the higher level is *Planner* agent, a deliberative agent whose main task is to receive the order from an external human operator to go to a certain room in its environment. This agent creates the appropriate navigation plan to safely navigate in the environment. The plan is made using a topological map of the environment that is available for the use of all the agents of the system.

When Planner agent creates the navigation plan, it is passed to the *Execution and Monitoring* layer. Its unique agent, *Monitor*, can develop several *skills* to achieve the sequence of sub-goals of the navigation plan. An *skill* is understood as the ability to accomplish a certain sub-goal and for this purpose it is used the activation of several behaviours either in a sequential or concurrent manner. These behaviours are implemented in the lower level, the *Execution* level, and *Monitor* agent activates them to accomplish a certain part of the plan while monitors the execution of these behaviours to check a possible error. It is also in charge of the localization of the robot during its movement on the environment.

The Execution level is divided into two levels: the *Behavioural* level and the *Hardware* level. In the Behavioural level, there are agents that implement behaviours. A *behaviour* can be seen as a relation between the inputs of the sensors and the actuators to achieve a desired goal. This concept regards to a lower level way of acting that directly relates environmental information received from sensor with the immediate actions that the robot takes. In this layer, *Navigation* agent implements several fuzzy behaviours that allow the robot to navigate safely in the environment. It uses information provided by range sensors in order to: avoid obstacles, approach to a door, cross a door and to navigate in a corridor among others abilities. There is also a *Vision* agent that implements visual behaviours to process the images captured from the environment. It can detect a desired landmark and move the PTU in order to fix it, indicating the straight direction to the door.

In the Hardware level there are agents that act as wrappers for the real hardware of the system. *Nomad200* agent establishes a communication via TCP/IP with a daemon running in the robot. Any agent that desires to send a command to the robot has to do it through this agent. The agent redirects the command in an appropriate format to the daemon allowing a concurrent use of the robot. The commands that can be sent allow to configure the speed and acceleration of the robot, to obtain the data from the ultrasound and infrared sensors and to move it. PTU agent allows the use of the PTU to the rest of the agent community. The main advantage of using this agent is the possibility of queuing the PTU movements in a remote way besides controlling the concurrent use of the PTU. The agent gives the service of moving the PTU and to inform about the exact position of both axes (pan and tilt) by message-passing. The camera has not been wrapped by an agent because the transference of the images on the net would speed down the work of the *Vision* agent. In the following sections all agents except for the hardware ones are explained in detail.

III. PLANNER AGENT

It is a deliberative agent that acts as a bridge between an human operator and the middle layer of the system. As we have previously commented we have identified two separated navigation tasks and we focus this work on the ability to navigate from one room to another into an office-like environment. When an human operator commands the robot to go to a certain room, this agent creates the navigation plan as a sequence of rooms and corridors to transverse and doors to cross. This sequence turns out to be sub-goals that are passed to Monitor agent which is able to develop several skills to accomplish them.

To create this navigation plan the agent makes use of a map of the environment that is in a shared representation structure available for all the agents of the system. Formally, the topological map consist in a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ is the set of N nodes, and $E = \{e_{ij} = (v_i, v_j),\ i = 1 \ldots N,\ j = 1 \ldots N\}$, is the set of M edges. The nodes of this graph correspond to distinguished places of the environment and the edges connect pair of this places. A fuzzy perception system [16] is able to classify the distinguished places according to its morphological characteristics in: corners (c), doors (d), hallways (h), end of corridor (ec) and a default object type corresponding to long irregular boundary (i). Then each edge of the graph represents either a wall, a corridor, an edge that crosses a door or a link between an irregular type node and any other kind of node, and it expresses a transition between two distinguished places. The map that can either be manually supplied or autonomously created in an exploration phase. The autonomous creation of the map is based on graph node saturation described in reference [17]. In Figure 3 there is the top view of a typical floor and in Figure 4 its corresponding topological map is depicted. Each door has an unique number in order to identify it in the map. This number is printed on the landmark placed beside the door and stored in the nodes that represent doors.

Using this map the agent can determine the path from an origin room to a destination one as a sequence of rooms, corridors and doors using the A^* algorithm. This is a high level of abstraction plan since it does not specify the movements of the robot in detail but a sequence of places that the robot has to reach. Lets imagine we wished our robot to go from $Room1$ to $Room2$ in the map shown in Figure 4. The plan would consist in the following path $Room1 \rightarrow d1 \rightarrow h1 \rightarrow h2 \rightarrow d2 \rightarrow Room2$. These sub-goals are: find the

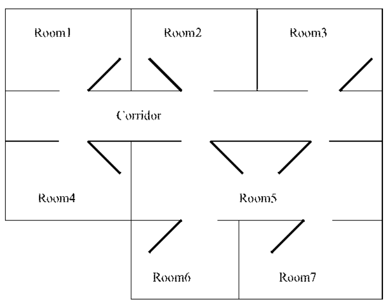

Fig. 3. Environment.

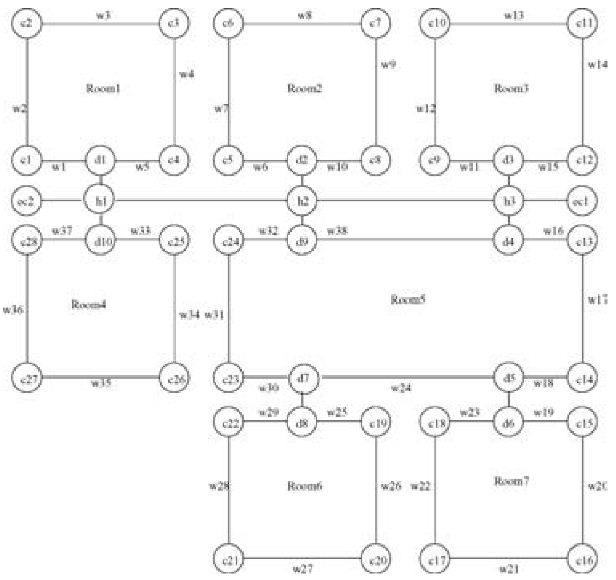

Fig. 4. Map of the environment.

door d1, cross it, travel along the corridor looking for door d2 and finally cross d2 to enter Room2.

It is possible that during the execution of the navigation plan an error occurs. In that case the Monitor agent fixes the error if it is possible. Otherwise, the agent informs about the error to the Planner agent which reports the incident to the human operator.

IV. MONITOR AGENT

This agent is mainly in charge of the execution of the appropriate skill to accomplish each part of the plan and also monitors the execution of the behaviours to check if the navigation plan is being achieved. As we explained above, a skill consists in the ability to achieve a certain sub-goal and for this purpose a concurrent or sequential activation of behaviours is used. Skills can be easily added to the system either incorporating new behaviours to the existing agents or new behavioural agents.

Planner agent passes the navigation plan to this agent that transform it in terms of the available skills. At the moment the plan is translated to the four possible skills: *find door in room*, *find door in corridor*, *approach to door* and *cross door*. Therefore the navigation plan explained above would be translated as: *Find door d1 in room* → *Approach door d1* → *Cross door d1* → *Find door d2 in corridor* → *Cross door d2*.

These are the four skills explained more in detail:

a) *Find door in room:* Every time the robot needs to find a door, the searching method depends on whether the robot is in a room or in a corridor. It is more usual to have enough visibility to distinguish the landmark placed beside the door when the robot is into a room than when it is in a corridor. Therefore we distinguish between the search of the landmark in a room and in a corridor. In any case, Monitor agent leads on Vision agent to detect the required landmark as well as on Navigation agent to move safely in its environment.

In order to find a door in a room, we have opted for searching the landmark with the robot stopped turning over itself (static search). The idea is that the robot turns the turret while the visual behaviour *Landmark detection* (explained in Section V with the rest of visual behaviours) searches the desired landmark in the images captured. Two conditions are required in order to achieve that, first of all, the landmark must be visible from its current location. And second, the distance from the robot to the landmark must allow to recognize it. Regarding the latter case, we have tested the vision system and adjusted it to work in typical indoor environment rooms. In our experimentation the vision system is able to identify the door at distances ranging from 1 meter to 5 meters. In case of failure the robot starts an exploration process using both visual and range information into the room moving parallel to the walls of the room while rounds it. The camera is pointed towards the wall that the robot follows and analyzes the images captured looking for landmarks. This skill is performed by the concurrent use of the fuzzy behaviours *Follow wall* and *Avoid obstacle* and the visual behaviour *Landmark detection*. All fuzzy behaviours are explained later in Section VI.

A key point to see a landmark is to set the camera inclination correctly to make it appears in the captured images. Lets call α the camera inclination angle, D the distance from the camera to the wall, h_m the distance from the landmark to the floor and h_c the distance from the camera to the floor. Figure 5 shows graphically the relation between these variables showing that α can be calculated using the Equation 1. To measure D we use the average of several values provided by the three frontal sonar sensors. Although it is an approximated method submitted to vagueness, it has been tested experimentally obtaining good results for the calculation of α.

$$\alpha = \mathrm{atan}\left(\frac{h_m - h_c}{D}\right). \tag{1}$$

b) *Find door in corridor:* Whenever the robot leaves a room and enters in a corridor, consults the map to know in what direction it has to go to reach the next room (left or right). The robot orientates its wheels to face the corridor in the correct direction and the turret to point towards the wall where the door is using the fuzzy behaviour *Orientate wheels & turret*. When the robot is correctly placed, a behaviour

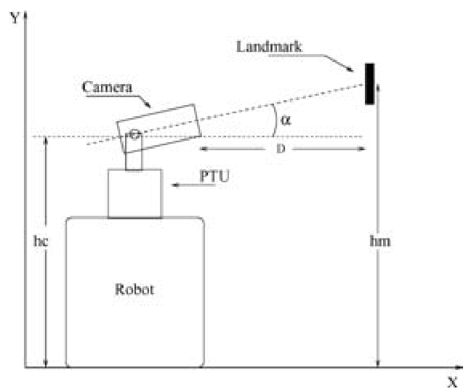

Fig. 5. Appropriate angle to detect the landmark.

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

that moves the robot along the corridor (*Follow corridor*) is activated to maintain the robot centered in the corridor while *Avoid obstacles* allows a safe navigation. At the same time, *Landmark detection* behaviour is activated searching for the landmarks in the appropriate wall.

c) *Approach to door:* Once the desired door is found, the robot moves towards it. The use of this skill is only necessary when the landmark has been located in a static search, since if it is located either during a search in movement or in a corridor, the robot is already placed near the door. To achieve this goal, the visual behaviour *Landmark fixation* is activated in order to keep the landmark in the field of vision indicating to the robot the straight way to the door. At the same time the fuzzy behaviours *Go to door* and *Avoid obstacle* are activated in order to lead the robot to the door without colliding with obstacles. It is possible that while approaching to the landmark the robot changes its straight way due to an obstacle. In that case, the Vision agent could loose the landmark from the field of vision loosing the appropriate way to the door when the obstacle is avoided. If so, a visual search using the behaviour *Find lost landmark* is done in the area where the landmark was found last time. In case of not finding it, the searching process in the room starts again.

During the approach, information about the distance from the camera to the landmark is provided by Vision agent. This information is used to decide when the robot is close enough to the door in order to develop the appropriate skill to cross it. The skill *Approach to door* stops when the robot is at an appropriate distance to ease the work of the skill that crosses the door.

d) *Cross door*: This skill allows the robot to cross a door when it is close enough. The first step is to place the robot in a favourable position to cross the door. For that purpose, *Center door* moves the robot to place it in front of the door using information gathered from the map and the range sensors. Then, two fuzzy behaviours are used: *Go through door* and *Avoid obstacle*. The first one creates the appropriate path to cross the door detecting the position of its frames while the second one avoids possible obstacles in the way.

Using these skills appropriately, the robot is able to go from its current location to a destination room. While the behaviours are running there is a bidirectional flow of data to inform about the status of each process. For example, the Vision agent would inform if the landmark that the robot is approaching to has been lost from the field of vision to make Monitor agent activate the behaviour *Find lost landmark*.

During the execution of the plan, Monitor agent keeps information about its position in the map to re-plan if it is necessary. This agent gets information about the doors that are passed in the way to the next door in the plan and this information is used to compute the position of the robot. In this way the robot is able to know in which room it is or in which node of the corridor is placed.

Monitor agent is also in charge of detecting possible errors during the execution of the navigation plan. The errors that can be detected by our system can be divided into recoverable and non-recoverable errors. Non-recoverable errors are those that can not be repaired by the system itself, therefore they are reported to the upper layer that informs to the human operator. These errors occurs, for example, when the robot does not reach the door while it is approaching to it because there is an unavoidable obstacle in its way or when a certain landmark is not found in a room. Nevertheless, there are errors that can be detected and recovered. If for example the search of the landmark is done in a corridor and it is detected that it has been passed the robot repeats the search process in the opposite direction.

V. VISION AGENT

Vision agent is in charge of detecting the desired landmark and keep it in the field of vision despite the movements of the robot towards it. The agent can detect the set of landmarks in a captured image and inform if a certain landmark is in it. When the desired landmark needed to accomplish the current part of the plan is identified, the agent can fixate it using the PTU indicating the straight way to the door. This agent can also measure the distance between the landmark and the camera indicating if the robot is close enough to the door to start the appropriate skill to cross it.

This agent implements three behaviours, the first one is called *Landmark detection* and consists in detecting if a certain landmark is present in the image captured at the current position of the robot. The second one is called *Landmark fixation*, it consists in keeping the landmark always in the field of vision (moving the PTU to center the landmark) despite the movements of the robot. Finally, the behaviour *Find lost landmark* searches for the landmark that was lost while the previous behaviour was running.

The landmark employed has been designed taking into account the following principles: it must be easy to distinguish from the environment, it must include information to uniquely identify the door, and it must be cheap and simple. The landmark finally developed is shown in Figure 6. It has an external black border that makes it easy to distinguish from the background and has some numbers inside to identify the door. The landmarks have been designed using a simple text editor and it fits in an A4 paper.

V.1. Landmark detection

The landmark detection is performed in two stages: the detection of the external border and then the determination of the number inside the landmark. The process starts when an image from the environment is captured. A segmentation process is performed to separate the objects that could be a landmark from the rest of the objects of the environment. An automatic thresholding method is needed due to the illumination changes that occurs in real environments. We have opted for the method proposed by Otsu[18] that selects a threshold value based on the illumination in order to minimize the entropy in the result image. This method has

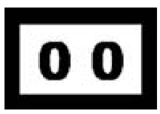

Fig. 6. Designed landmark.

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

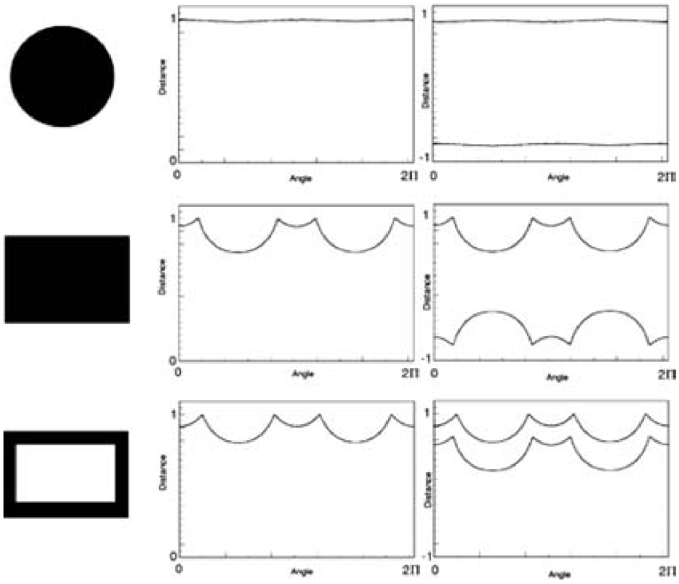

Fig. 7. Examples of the signature and double signature applied to three shapes.

shown to give good results in our experiments and is not time-consuming.

The next step consists in detecting the external border of the landmark. Hence, every object present in the binarized image is analyzed to know if is a landmark or not. The first attempt was to use the *signature* as descriptor.[19] It is a function $f(\phi)$ that represents the contour measuring the distance from the center of the object to the border point that is in a line traced with a certain angle ϕ ranging in $[0, 2\pi]$. However, signature does not allow to detect holes in objects so all rectangular objects have a similar representation. Instead of signature, we have developed a variant of the method that has been denominated *double signature*. It consists in tracing lines form the center to the external border of the object for angles ϕ in the range $[0, 2\pi]$. Each line is scanned starting from the most external border point that is in the angle ϕ. The scan is performed in direction to the center, and the distances from the center to the two first border points found are measured. The distance is considered negative if the point is found after pass the center in the scanning process. In order to make the representation invariant to scale all the distances are normalized to the maximum one. Figure 7 shows three shapes (left column) and either the signature (center column) and double signature (right column) of them are represented. This method brings two main advantages. First, it has been proved to be more robust in our experiments for detecting the landmarks using several classification methods, and second it allows the description of holes in the objects that it represents.

Double signature distances are stored in an array to represent the objects that appear in the image after segmentation. In order to learn the double signature of the rectangular black border a neural network has been used. The network has been trained using the back-propagation algorithm and the best network found has 76 inputs a hidden layer with 8 neurons and 2 outputs. The training and validation set was created using 135 images where the landmark appears from different points of view. From each image a positive example is taken (the landmark in the image) and the rest of objects are considered negative examples. In this way the total number of patterns extracted is 5281, using the 70% for training and the rest for validation. The neural network obtains a 100% success percentage over the training set and 99.65% in validation.

Once the landmark has been found it is necessary to determine the number that is inside. To do this we analyze each object (greater than a certain number of pixels) into the detected rectangular border and classify it. Each digit is adjusted to a 16×16 image and for each one the following features are extracted:

a. *Horizontal and vertical Laplacian*. It consists in creating a function with the number of pixels occupied by the digit in every row and column, then obtaining the Laplacian of this function and escalating it to the range $[-1, 1]$.
b. *Block Sum*. Consist in the creation of 2×2 blocks and counting the number of pixels occupied by the digit in each block. Finally, the value is normalized dividing by 4.

A vector of 74 elements describing the digit is created using these measures. A neural network has been created and trained to classify the digits. In our experiments, digits have always been correctly classified.

V.2. Landmark fixation

Once the landmark has been located, it is necessary to fixate it. The objective is to have the landmark always placed into the next image to show the straight direction to the door. This behaviour uses the PTU to center the landmark in the image. The unique information needed to center the landmark in the field of vision is the pixel distance from the center of the landmark in the image to the center of the image. Then it is necessary to establish the correspondence that leads the PTU to center the landmark in the image.

We wish to obtain a fast lace of control to avoid loosing the landmark. We assume that the nearest landmark to the center in the current image is the desired landmark. It is reasonable if the landmark has not been loosed in a previous loop and the control lace is fast enough. In this case the computing time can be reduced by detecting only the external border and not processing the digits. Smaller images can be used for this process because the precision required is less important.

In order to determine how many degrees the PTU must be turn in both axis two regression lines has been calculated. They indicate the necessary angle increment based on the pixel distance from the center of the image to the pixel that it is desired to be centered. The lines calculated based on real values are:

$$\text{pan\_degrees} = -25.99 \cdot \frac{2\text{IncPixX}}{\text{Width}} + 0.05$$

$$\text{tilt\_degrees} = 17.63 \cdot \frac{2\text{IncPixY}}{\text{Height}} + 0.29$$

Where *IncPixX, IncPixY* represent the pixels distance from the center of the landmark in the image to the center of the image in both axis X, Y. *Width* and *Height* are the width and height of the image respectively.

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

V.2.1. Distance measurement. The monitor agent needs to know the distance between the robot and the landmark to determine if it is close enough to the door. If so, Monitor activates the appropriate skill to cross the door and this agent stops the fixation process. It is possible to measure the distance using the ultrasound, but it submits the system to the typical problems of these sensors, like false reflections due to the furniture near the door. Eventually, we have used vision to perform the measures because it have provided better results than ultrasound in our experimentation.

Assuming that the size of the landmark is known and fixed for all landmarks, we can approach the distance to the door measuring the size in pixels of the vertical projection of the landmark in the image. The closer the mark is to the robot, the bigger the vertical projection is, and *vice versa*. The horizontal projection is not so appropriate because is more affected by perspective deformations due to the wide range of position in which it can be seen. While the range of angles under which the landmark can be seen in the horizontal axis is wide (when the robot rounds the door), in the vertical axis it is limited by the height of the camera and the height at which the landmark is placed. We have extracted a regression curve by least-squares using real values to approximate the distance D. The curve obtained is:

$$D = 763.02 - 6652.21x + 23403.03x^2 - 29177.50x^3$$

Where $x = \frac{\text{NoPixels}}{\text{Height}}$, *NoPixels* is the number of pixels of the vertical projection of the landmark in the image and *Height* the height of the image. Therefore, x is in the range [0,1], so the curve is valid for different image sizes.

V.3. Find lost landmark

This behaviour is activated when the landmark is lost from the field of vision because of the movements of the robot. In this case, the robot looses the straight direction towards the landmark and, instead of performing a full search in the room, it is preferable to search the landmark in the direction where the robot was looking at. For that, the agent turns the PTU to the left and right side and performs a visual search of it in this area. This process is faster than a full search because the area covered is smaller and usually succeed. But if it does not succeed, a complete visual search in the room is ordered by the Monitor agent.

VI. NAVIGATION AGENT

As commented in Section II.2, this agent is in charge of carrying out the navigation of the robot safely in the environment. It relates the perception of the range sensors to the orders which must be sent to the actuators according to the objective that must be executed in each moment. For this reason, it uses different behaviours designed using techniques taken from the Fuzzy Control Theory.[20]

The use of these fuzzy behaviours allows the robot to control the actuators despite the presence of noise in the data sensors that cause inaccuracy in the measures and uncertainty about the information used. Below, we explain how the behaviour to guide the robot to the door has been designed whereas for the rest of the behaviours only their function is commented.

a) *Go to door.* This behaviour permits the robot to reach the landmark of the indicated door orientating the wheels and turret in the direction that is indicated by the PTU. When the behaviour is activated by the Monitor agent, the desired landmark has already been found and the Vision agent makes the PTU point towards it. It is necessary to indicate that the hardware of the robot allows to move the turret independently from the wheels. It gives the robot the possibility of maintaining the turret orientated towards the door when the wheels are turn in order to avoid some obstacle. In this way the work of the Vision agent is helped because less movements of the PTU are required when avoiding an obstacle. Therefore, two fuzzy rule set have designed. One to orientate the turret and another to set the orientation of the wheels.

- Rule set for orientating the turret. The goal is to keep the turret oriented towards the landmark to ease the work of Vision agent. For this reason, according to the angle of the PTU and the present situation of the turret, the angle that the turret must be turned to remain lined up with the PTU is calculated. This angle, denoted as *Angle Move Turret* (*AMT*), is used as input for the rules to control the *Turret Rotation Speed* (*TRS*). Figure 8 shows the labels of *AMT* and Figure 9 shows the labels for *TRS*. In Table I(a) the rule set to achieve the desired goal is shown.
- Rule set to orientate the wheels. The idea is similar to the previous one, to maintain the wheels of the robot lined up with the PTU to lead the robot towards the landmark. In this case the variables to control are the *Wheels Rotation Speed* (*WRS*) and the *Translation Speed* (*TS*) of the wheels. Thus, the necessary angle to line up the PTU and the wheels of the robot, *Angle Move Wheels (AMW)*, is calculated and used for controlling the output variables using an appropriate rule set. The labels of *WRS*

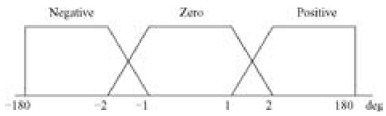

Fig. 8. Labels of variable AMT.

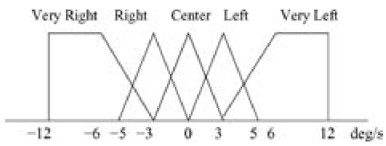

Fig. 9. Labels of variable TRS.

Table I. Rules set to control the variables (a) TRS (b) WRS (c) ΔTS.

A M T	TRS
NEG	Right
ZE	Center
POS	Left

(a)

A M W	WRS
NEG	Right
ZE	Center
POS	Left

(b)

A M W	ΔTS
NEG	Negative
ZE	Positive
POS	Negative

(c)

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

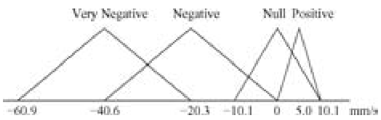

Fig. 10. Labels of variable ΔTS.

and *AMW* are similar to the labels of *TRS* and *AMT*, respectively, and Table I(b) shows the rule set for this variable. In order to ease the alignment, another variable (ΔTS) is used to adjust the variable *TS*. ΔTS is added to *TS* making the robot to decrease its speed up to a certain limit when it is not lined up with the PTU and vice versa. The labels for this variable are shown in Figure 10 and the rule set for this variable is in Table I(c).

b) *Avoid obstacles*. This behaviour allows the robot to avoid an obstacle that is placed in its way. The input variables that have been taken into account are: *Frontal Distance*, *Right-Left Distance*, *Right-Left Difference Distance* (all of them detected by ultrasound) and *TS*. The variables to control are *WRS* and *TS*. The idea that leads the design of this behaviour consists in the fact that if the obstacle is not in front of the robot, it tends to turn a little either to left or right, depending of what is safer. When the obstacle is just in front of the robot, or when it has approached too much to it, the robot turns towards a safe place. In any case, when an obstacle is detected the speed decreases.

c) *Follow wall*. The objective of this behaviour is to allow the robot to move around a room following the wall outline. The input variables for the behaviour are *Wall Distance* and *Incidence Angle* to the wall while the output variable is *WRS*.

d) *Follow corridor*. This behaviour allows the robot to go along a corridor, moving forward in the center of it. The input variables are *Right-Left Distance Difference* in regard to the corridor walls and *Right-Left Incidence Angle Difference*. The aim is to control *WRS* to center the robot as well as *TS* that increases if the robot is in the center of the corridor and decreases otherwise.

f) *Center door*. This behaviour places the robot in a favorable position to ease the work of *Go through door* behaviour. Crossing a door is a difficult task so the behaviour places the robot in front of the door and orientates it towards the gap to make the trajectory to cross the door as straight as possible. Some times it is possible to detect the gap of the door using range sensor information. In this case the behaviour can directly orientate the robot to face the door. In some other cases the gap can not be detected using range sensor information and the map must be consulted. It usually happens when the door is in a corridor because the landmark is found either before or after passing the gap of the door. This agent uses as input variables to detect the gap the *Frontal-Lateral Difference Distance* and the output variables are *TS*, *WRS* and *TRS*. If the robot is in a corridor consults the map to determinate if it has to move either forward of backward to find the gap of the door. The inputs variables in this case are the *Frontal-Lateral Difference Distance* to detect the gap and the outputs are *TS* and *WRS*.

g) *Go through door*. It allows the robot to go trough a door to enter or leave a room. A temporary model of the door has been used with the aim of calculating a straight line that crosses through the gap of the door. The robot takes as reference this trajectory but because this process is not free of noise, it is possible that the trajectory must be readjusted according to the success or failure of the behaviour. The input variables are: *Distance to trajectory* calculated and *Deviation Angle* in relation to the trajectory. The variable *WRS* has been considered as output variable while *TS* decreases to very low values.

h) *Orientate wheels & turret*. When the robot has to navigate in a corridor it is necessary to orientate the wheels to face it appropriately. Likewise the turret of the robot is oriented to point towards the wall in which the desired landmark is placed using information from the topological map. The inputs are *Angles Move Wheels & Turret* that show how much the robot must be turned to reach the desired position, and the output variables are *WRS* and the Turret Rotation Speed *TRS*.

Further details about the design methodology of the fuzzy behaviours can be seen in reference [16].

VII. EXPERIMENTAL RESULTS

This multi-agent system is structured in three levels. It has been implemented and tested using a mobile robot Nomad 200 modernized with a vision system, and placed in the real office-like environment represented in Figure 11. We explain the concurrent running of the whole system with a real example. The robot is initially situated in room 0 and has to get room 2, traveling through a corridor. The first step is to supply the robot with the initial and target point. With that information the Planner agent calculates the following path according to the topological map : Room $0 \rightarrow d0 \rightarrow h2 \rightarrow h3 \rightarrow h4 \rightarrow d2 \rightarrow$ Room 2. These steps represent sub-goals to accomplish in order to achieve the main goal and Monitor translates the plan into the possible skills as : *Find door* $d0$ *in room* $\rightarrow$ *Approach door* $d0$ $\rightarrow$ *Cross door* $d0$ $\rightarrow$ *Find door* $d2$ *in corridor* $\rightarrow$ *Cross door* $d2$. As it can be appreciated, it is a high level of abstraction plan where no trajectory is specified to the robot, but rather a set of sub-goals to reach. The plan is executed in the stages explained below.

a) *Find door* $d0$ *in room*: According to the path previously obtained, the robot searches for the landmark that identifies the door $d0$ by means of the static search. Figure 12 shows two images obtained by the robot in the static search process. In the second image of this Figure the landmark with the digits 00 appears on the left side of the door. In another situation, the landmark could be hidden by an obstacle, and thus be invisible to the robot. In this case, after completing the static search without success (after a 360 degrees turn) the robot would start a search moving parallel to the walls of the room avoiding possible obstacles. Figure 13 shows the path followed by the robot in this situation.

b) *Approach door* $d0$: If the robot finds the landmark 00 in the static search, it approaches to the landmark orientating the wheels and the turret in the direction indicated by the

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

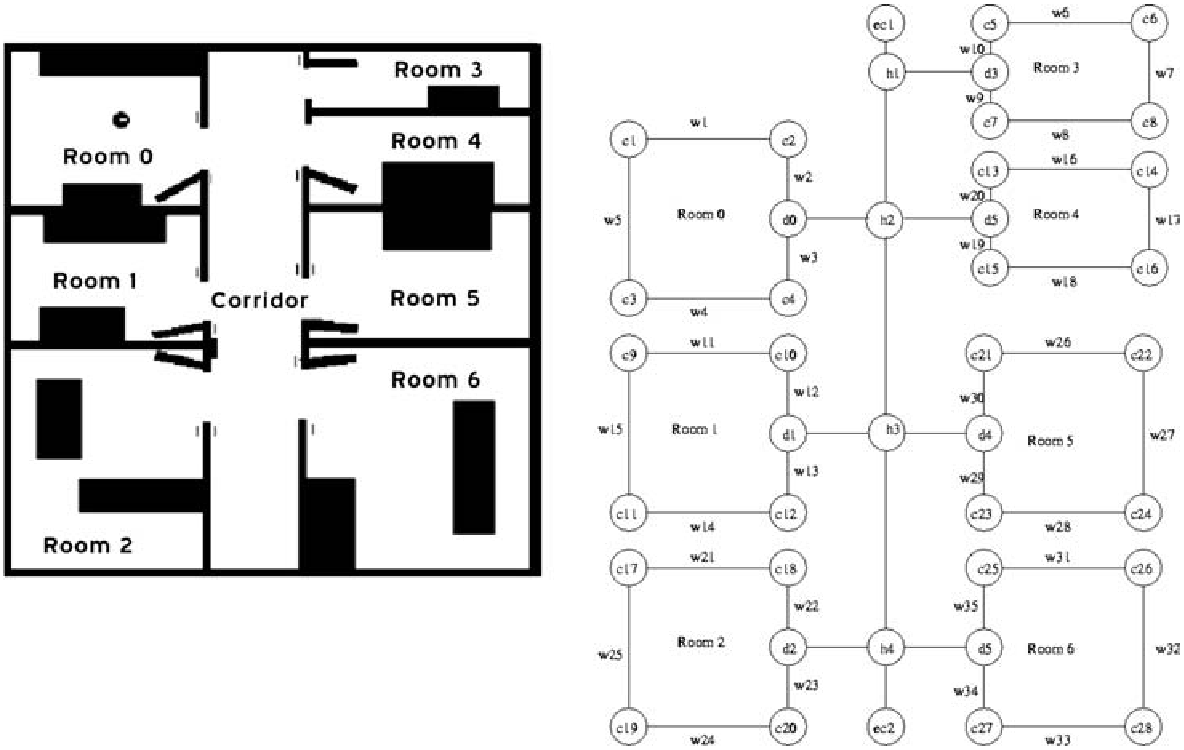

Fig. 11. Rooms map and topological map of the real environment.

Fig. 12. Static search of the landmark 00.

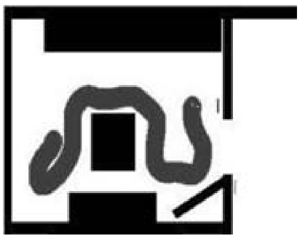

Fig. 13. Search in movement of the landmark 00.

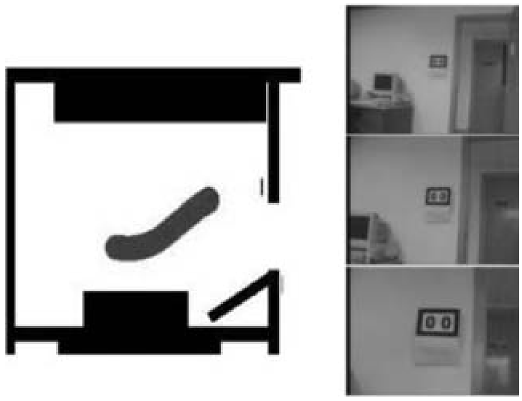

Fig. 14. Approaching to the landmark 00.

PTU. In Figure 14 there is shown the path followed by the robot while approaching to the landmark as well as several images taken in the process. This phase finishes when the robot is at an appropriate distance from the door. It must be noticed that this phase does not take place when the landmark is located by means of the process of searching in movement, since in this case the robot approaches the landmark in the searching process.

c) *Cross door d*0: When the robot is placed near the door, *Monitor* agent performs the skill *Cross door*. At first, the robot detects the gap of the door and orientates itself appropriately (using the behaviour *Center door*) to ease the work of the behaviour *Go through door*. The latter behaviour calculates the appropriate path to cross the door and moves the robot through the gap. The path of the robot and some images taken while the robot crosses the door are shown in Figure 15.

d) *Find door d*2 *in corridor*: Once the door has been crossed and the robot is in the corridor, it must navigate lined up in the center to find the destination room. For this reason, in a first phase the robot orientates the wheels to face the corridor and the turret to facilitate the visual search of the destination door. Then the appropriate behaviours move the robot in the center of the corridor avoiding possible obstacles while images of the wall are captured and analyzed to detect landmark 02. Figure 16 shows how the robot searches and finds the landmark of the door *d*2 navigating along the corridor.

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

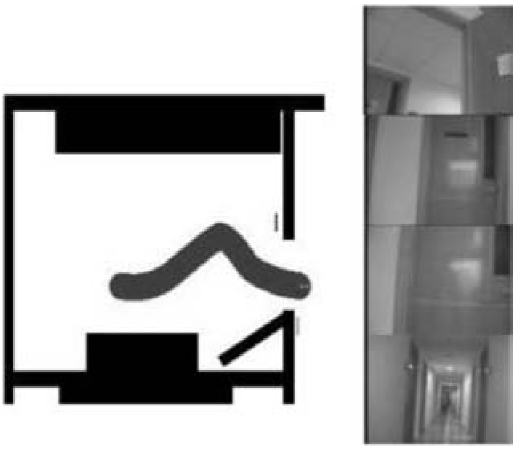

Fig. 15. Crossing the door *d*0.

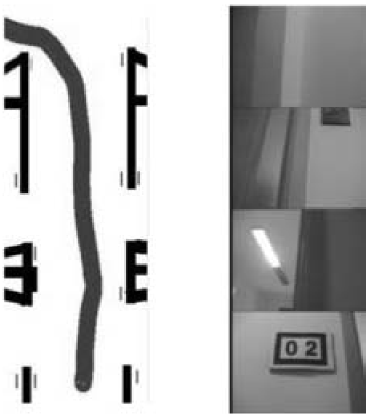

Fig. 16. Along the corridor searching for door *d*2.

Fig. 17. Crossing door *d*2.

e) *Cross door d*2: The skill *Cross door* is executed again and the robot achieve the navigation plan. When the landmark is found the robot moves backwards to place itself in front of the door in order to ease the work of the behaviour that crosses it. The path that the robot has followed and some images taken in the process are shown in Figure 17.

The images shown in the previous figures have been taken by the robot in a real execution of the system. This experiments and others have been filmed and can be downloaded at http://decsai.ugr.es/~eaguirre/research/Videos.htm

VIII. CONCLUSIONS

In this work we have presented a multi-agent system that employs both visual information and range information to control the navigation task of a mobile robot in office-like environments. The system is organized in a three-layer architecture consisting of several agents that share the responsibility to achieve the navigation task. This is performed by means of four skills: a) detect doors in rooms, b) detect doors in corridors, c) lead the robot towards a door and d) cross a door. The system is flexible to the addition of new agents, skills and behaviours and it can be adapted to other types of robots because of the standard hardware employed in this work.

The environment is represented using a topological map where nodes represent distinguished places and arcs join them. Besides each door an artificial landmark has been placed consisting in a rectangular border with digits inside to aid the robot to identify the door. The landmarks designed are easy to make with a normal text editor fitting into an A4 paper and their placement can remain discreetly in the environment. The landmarks and the digits are recognized using trained neuronal networks achieving high success rates.

A high level of abstraction agent generates the plan and passes it to the middle layer that translate it into a sequence of available skills. A visual agent that uses fast algorithm for image processing is able to maintain the camera focussed on the landmark using the PTU to indicate the straight direction to the door. A navigation agent that uses range information implements fuzzy behaviours to manage the underlying vagueness and uncertainty and allows to move the robot safely in the environment. It can guide the robot towards the landmark of the door, avoid the obstacles, follow walls, cross doors and move the robot along a corridor. The experiments carried out in our office-like environment show that the system is able to safely navigate and accomplish the desired navigation plan.

In future work we plan to replace the Vision agent for another that directly detects doors based on its shape and, thus, remove the artificial landmarks.

Acknowledgement

This work has been partially supported by the MCYT under Project TIC2003-04900.

References

1. D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," *Proc. of the Twelfth National Conf. on AI (AAAI-94)*, Menlo Park, Calif. (1994) pp. 979–984.
2. A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer Magazine, Special Issue on Autonomous Intelligent Machines* **22**(6), 46–57 (1989).
3. S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence* **99**(1), 21–71 (1998).
4. B. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems* **8**, 47–63 (1991).
5. R. C. Arkin, *Behavior-Based Robotics* (The MIT Press, 1998).

https://doi.org/10.1017/S0263574704001390 Published online by Cambridge University Press

6. R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation* **RA-2**, 14–23 (1986).
7. M. Mataric, "Integration of representation into goal-driven behavior-based robots," *IEEE Transactions on Robotics and Automation* **8**(3), 304–312 (1992).
8. E. Aguirre and A. González, "Integrating fuzzy topological maps and fuzzy geometric maps for behavior-based robots," *International Journal of Intelligent Systems* **17**(3) 333–368 (2002).
9. G. N. Desouza and A. C Kak, "Vision for mobile robot navigation: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 237– 267 (2002).
10. Hao Li and S. X. Yang, "A behavior-based mobile robot with a visual landmark-recognition system," *IEEE/ASME Transactions on Mechatronics* **8**, 390–400 (2003).
11. R. Katsuki, J. Ota, T. Mizuta, T. Kito, T. Arai, T. Ueyama, and T. Nishiyama, "Design of an artificial mark to determine 3d pose by monocular vision," *IEEE International Conference on Robotics and Automation, 2003. Proceedings ICRA '03* (2003) **Vol. 1**(14–19), pp. 995–1000.
12. D. Scharstein and A. J. Briggs, "Real-time recognition of self-similar landmarks," *Image and Vision Computing* **19**, 763–772 (2001).
13. K. Tashiro, J. Ota, Y. C. Lin, and T. Arai, Design of the optimal arrangement of artificial landmarks, *1995 IEEE International Conference on Robotics and Automation* (1995) pp. 407–413.
14. A. Saffiotti, K. Konolige, and E. Ruspini, "A multivalued logic approach to integrating planning and control," *Artificial Intelligence* **76**, 481–526 (1995).
15. E. Aguirre and A. González, "Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion," *International Journal of Approximate Reasoning* **25**, 255–289 (2000).
16. E. Aguirre and A. González, "A fuzzy perceptual model for ultrasound sensors applied to intelligent navigation of mobile robots," *Applied Intelligence* **19**(3), 171–187 (2003).
17. P. Panaite and A. Pecl, "Exploring unknown undirected graphs," *Journal of Algorihtms* **33**(2), 281–295 (1999).
18. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions On-Systems, Man, and Cybernetics*, **SMC-9**(1), 66–69 (1979).
19. R. González, *Digital Image Processing* (Addison-Wesley Iberoamericana, SA, 1996).
20. D. Driankov, H. Hellerdoorn, and M. Reinfrank, *An Introduction to Fuzzy Control* (Kluwer Academic Publishers, 1993).