World Academy of Science, Engineering and Technology
International Journal of Mechanical and Materials Engineering
Vol:2, No:10, 2008

Acquiring Contour Following Behaviour in Robotics through Q-Learning and Image-based States

Carlos V. Regueiro, José E. Domenech, Roberto Iglesias, and José L. Correa

Abstract**—** In this work a visual and reactive contour following behaviour is learned by reinforcement. With artificial vision the environment is perceived in 3D, and it is possible to avoid obstacles that are invisible to other sensors that are more common in mobile robotics. Reinforcement learning reduces the need for intervention in behaviour design, and simplifies its adjustment to the environment, the robot and the task. In order to facilitate its generalisation to other behaviours and to reduce the role of the designer, we propose a regular image-based codification of states. Even though this is much more difficult, our implementation converges and is robust. Results are presented with a Pioneer 2 AT on a Gazebo 3D simulator.

Keywords**—** Image-based State Codification, Mobile Robotics, Reinforcement Learning, Visual Behaviour.

I. Introduction

THE design and implementation of reactive behaviours for the control of autonomous mobile robots has been shown to be one of the most efficient ways of carrying out low level tasks. These require a very short response time and continuous interaction with the environment, which is almost totally unknown, complex and dynamic [1]. Thus arises the challenge of specifying which is the most suitable set of behaviours to implement for a specific robot-environment-task triad [2] and how each one should be implemented.

In this aspect, the application of soft-computing techniques has arisen naturally in the development of different behaviours: artificial neural networks [3], [4], genetic algorithms [5], fuzzy logic [6], etc. One of the most promising is the reinforcement learning (RL) [7], [8], one of the principal advantages of which is that it minimises interaction with the designer, since only the set of states and actions and the reinforcement has to be established. There is no need to identify all the situations in which the robot may find itself, nor the action to be implemented in each of them. It only needs to be stated whether the result of the action is acceptable or not.

Along this line a number of different behaviours have already been learned with sensors that are typical of mobile robotics, principally ultrasound [9], [10], [11], [12]. Nevertheless, one important drawback of these behaviours is the type of sensor that they use, as they generally only perceive obstacles that are located on a plane that is parallel to the ground.

To avoid this we can employ other types of sensors, such as, among others, 3-D lasers or artificial vision. In the former case, the main problem is that the equipment is highly expensive, bulky, heavy and energy-consuming. The drawbacks for artificial vision are the dependence on lighting and the texture of the different components that make up the environment, and the computational cost of processing the information generated. On the upside, they are small and cheap.

This work describes the design and implementation of indoor contour following behaviour using a single camera, with the aim of studying the feasibility of the project and enabling a simple, economical implementation. The results obtained are generalisable, with the possible exception of the discrimination between floor and obstacles. This perception would be more robust and efficient if in-depth information were used (e.g., stereo vision).

We now comment on related work and go on to describe the Q-learning algorithm and its application to contour following behaviour. We will then show the experimental results. Lastly, we finish off with a conclusions and future work section.

II. Related Work

Only a small number of studies have used vision as the principal sensorial input for RL in a mobile robot. This is probably due to the high cost associated with processing visual information [4]. In some works, visual behaviours are learned by reinforcement that are similar to contour following (e.g. servoing and wandering), but which are simpler as there is no need to maintain a distance from the contour the robot is following. Gaskett et al. [13] use an improved version of Q-learning ("Advantage Learning") which handles continuous states and actions thanks to neural networks.

Another implementation of visual servoing behaviour can be found in [14]. The algorithm used is another variant of Q-learning that permits real-time learning. Unlike in the present work, the state of the agent is defined by the position of the camera (inclination and angle of turn) focused on the objective, and not by the image. Thus, active vision is required, along with a perfect identification with the objective, which is not always possible. It is also difficult to use the same system to implement more than one behaviour simultaneously.

A similar, but more complete, general approximation were taken in [15]. This system learn basic behaviours (watching and orientation) for controlling a pan-tilt unit and the robot, and combine them to obtain complex ones (approach). Nevertheless, they need to detect and identify the objective, which is very difficult with a contour. Again, states were no defined directly by the image.

Ruiz et al. [16] have implemented two visual behaviours, one being wall following. Their approach is based on the

Carlos V. Regueiro and José E. Domenech are with the Deparment of Electronics and Systems, University of A Coruña, Spain.

Roberto Iglesias and José L. Correa are with the Deparment of Electronics and Computer Science, University of Santiago de Compostela, Spain.

detection of straight segments in the image, which to a certain degree limits its mode of use. Moreover, control is totally heuristic, with no type of learning. Our approach is more general and is not conditioned by the type of wall.

Nehmzow [17] has succeeded in making the robot learn two visual behaviours with neural networks. The first consists of following walls and corridors, avoiding obstacles with supervised learning, indicating the best action to be implemented at each instant. This type of training is especially tedious for the designer, and the result is neither robust nor generalisable. The second behaviour consists of detecting boxes in a static image by means of unsupervised leaning, and subsequently approaching them. The system learns, but without robustness.

Some works are based on an omni-directional (catadioptric) camera. Visual path following and corridor following behaviours were implemented in [18] for topological navigation. First, discrimination between floor and walls was made and the detected points were adjusted to an ideal corridor. Finally, an heuristic control was used.

III. Reinforcement Learning

Reinforcement learning (RL) [19], [20] is based on the use of a qualification (reinforcement) of the agent's actions by the environment. The reinforcement does not indicate the correct action (supervised learning), only whether it has been satisfactory or not, and is not usually immediate in time. Usually, the situations of the agent are codified into discrete states (s) in which various actions (a) can be implemented (may be different for each state). Learning consists of approximating a quality function $Q(s, a)$. The optimal action in each state is the one that maximises its evaluations. The algorithm chosen for this work is *Q-learning*, due to its simplicity and easy implementation. The updating equation for the valuations is:

$$\Delta Q(s_t, a_t) = \alpha[r_t + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \ , \quad (1)$$

where α is the learning coefficient and γ is the reinforcement discount coefficient. Initial Q-values are between -0.9 and -1.0.

One drawback of *Q-learning* is the need to strike a balance between exploration and exploitation [19]. In order to do so, the *Softmax* algorithm [11] has been applied, where the probability of taking the action a_i in the state s at time t is:

$$Pr(s, a_i) = \frac{e^{Q_t(s,a_i)/T_s}}{\sum_{j=1}^{n} e^{Q_t(s,a_j)/T_s}} \ , \quad (2)$$

where $\{a_1, ..., a_n\}$ is the set of possible actions in the state s and T_s is the temperature associated to state s.

With temperature it is possible to regulate the probability distribution between actions, and thus, the balance between exploration and exploitation. Initially we start from a high temperature (greater exploration of actions) and this is progressively reduced throughout the learning in order to principally select those action with the best evaluation. As the frequency with which the different states appear is highly variable, the choice was made to regulate their exploration/exploitation balance (reducing temperature) individually in accordance with the following equation:

$$T(t) \quad = \quad \begin{cases} T_0 \, e^{-\frac{t}{t_k} ln \frac{T_0}{k}} & si \ t \leq t_k, \\ k & si \ t > t_k, \end{cases} \quad (3)$$

where t is the number of times the current state has appeared, T_0 is the initial temperature, k is the minimum temperature (the state does not explore more actions) and t_k is the number of appearance of the state that are required for the temperature to reach k.

With this formulation, it is possible to directly specify how many appearances are required to neutralise the exploration for a state. Table I shows a summary of the values used for the parameters of the Q-learning algorithm.

TABLE I
Values used in this work for the Q-learning algorithm

Par.	Description	Valour
α	Learning coefficient	0,2
γ	Reinforcement discount coefficient	0,99
T_0	Initial temperature	0,07
k	Lower temperature limit	0,009
t_k	State exploration limit	4.000

IV. Contour Following Behaviour

It has been shown that contour following behaviour is one of the most useful when robots need to move reactively and safely through their environment [1]. One of its advantages is that it only uses local information, and it makes use of the topological structure of the environment. The selection of camera and its placement on the robot are important aspects. The angle of vision of a normal camera is not sufficient to cover the robot's immediate environment. A wide-angle lens would permit a greater field of vision, but would excessively distort the image, and resolution would be lost. Using various cameras complicates the system unnecessarily.

On a reactive behaviour and with RL, each state must content all the information to select the next action. Therefore, the contour needs to be in all images. After various tests the camera (figure 1) was placed some 25 cm above the robot, inclined 35 degrees towards the floor, and turned 40 degrees to the right (values that are very similar to those used by Nehmzow [17]).

A. State Space

As was to be expected, the definition of the state space was critical in the development of this work, since it has to respond to sometimes conflicting criteria. On one hand, the space must be small, as convergence time in RL increases exponentially with the number of states. On the other hand, "perceptual aliasing" must be avoided; that is, the system should not classify two situations in which the robot must execute very different commands in the same state. In this case, the result would be a high level of instability in the learning, and most probably it would not converge.

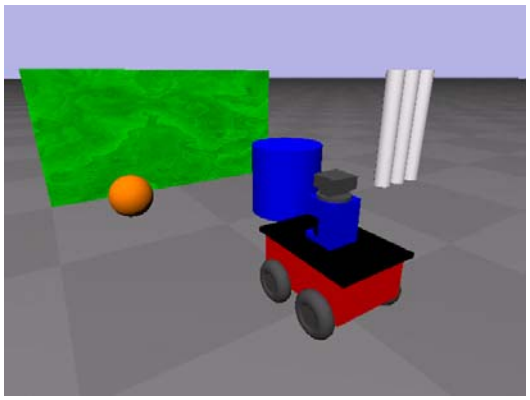

Fig. 1 Position of the camera for contour following behaviour

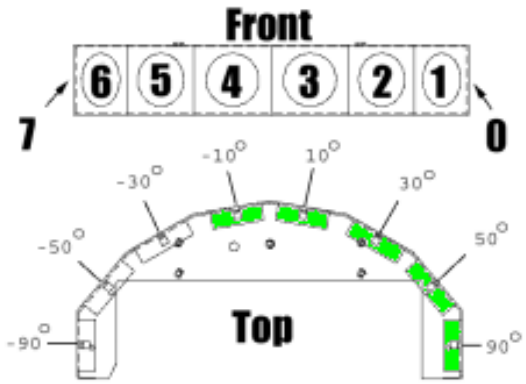

Fig. 3 Ultrasound sensors of the P2 AT selected to define reward

Lastly, due to the quantity of information generated by a camera, even at a low resolution such as 320x240 pixels (figure 2 (left)), the image needs to be processed in order to reduce the amount of pertinent information [17]. In order to resolve these problems, a simple, computationally efficient methodology has been employed that can be run in real time on a mobile robot. The approach may be limited, but it is easily generalisable.

Firstly, the image is processed with a Sobel edge enhancement filter (figure 2 (center)) to highlight the pertinent information: obstacles (positive and negative) on the floor. This floor detection process is highly sensitive to changes in lighting and textures; nevertheless, it can be improved in different ways: with stereo vision [21] (information on depth), by calibrating the camera to detect the ground plane [22], [23], or by applying techniques of Machine Learning for ground boundary detection [24], [25], [26].

Secondly, the image is divided into a grid made up of 8 rows and 4 columns (figure 2 (right)) for codification. A cell is considered occupied if the percentage of edge pixels reaches a given threshold. This step is essential for avoiding "perceptual aliasing". Thus defined, the state space is enormous (2^{32}), and in order to reduce it, it is supposed that if a cell in one of the columns is occupied, all those cells above it are occupied too (figure 2 (right)). Hence the number of possible states is $(8+1)^4$; i.e. 6561. The state space may be further reduced, but drastic modifications would be needed in the codification, which would be difficult to justify.

B. Action Space

In order to simplify the learning of the task, it has been supposed that the robot's linear velocity is constant (30 cm/s) and that only the angular velocity need be learned for each state. One constraint of Q-learning is that actions must be discrete. Hence, the action space chosen is:

$$\omega = \{-0,3, -0,1, +0,1, +0,3\} \quad \mathrm{rad/s}. \tag{4}$$

C. Definition of Reinforcement

A simple and intuitive definition of reinforcement has been sought, as we believe that it is one of the main advantages of this type of algorithm. Reinforcement indicates *only* those situations in which it is highly probable that the robot has ceased to correctly implement the task (i.e., the robot is excessively far from a contour), or has definitively carried it out badly (the robot collides with an element in its environment or is excessively close to one).

The defined reward is always negative (-1.0), spurious in time and has two different components:

1) All cells on the right column are free (i.e., no contour is detected on the image).
2) Selected ultrasound sensors (see figure 3) detect an obstacle at 20 cm or less (i.e., very close to the robot).

V. Experimental Results

For reasons of safety, and to accelerate training, the learning phase was carried out on a simulator. The Player/Stage/Gazebo was chosen as it is highly generalised, it supports the Pioneer 2 AT mobile robot, and simulates in 3-D (Gazebo). The environment used is shown in figure 4.

Contour following behaviour belongs to the class of continuous tasks [19] which persist over time. This means that they are not divided naturally into episodes, as would be desirable for application of a reinforcement algorithm. In order to avoid this drawback a reinforcement is generated after each serious system error (collision or excessive distancing) and the robot is returned to the initial position of the training.

Figure 5 shows the reinforcement received during the learning phase. Each point represents the reinforcement accumulated in the previous 400 learning cycles. Thus, the first point indicates that the reinforcement received since the onset of the experiment up until 400 cycles has been -55, and so forth. The learned Q-values are stored for their subsequent testing.

The results of the test phase are shown in figure 6: the number of cycles during which the robot has executed the learned behaviour before committing a serious error. In this phase the optimal policy is applied; i.e. the action that is best valued for each state is carried out.

The convergence criterion for the learning is that three consecutive sets of Q-values should have been accurately

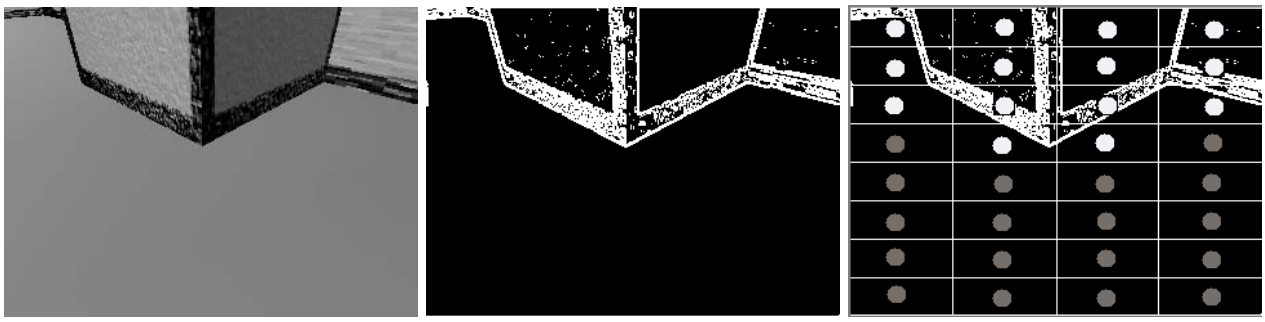

Fig. 2 Determination of state with an image: (left) original image; (center) edge pixels (Sobel filter); and (right) final codified state (showing the codification grid and the free and occupied cells)

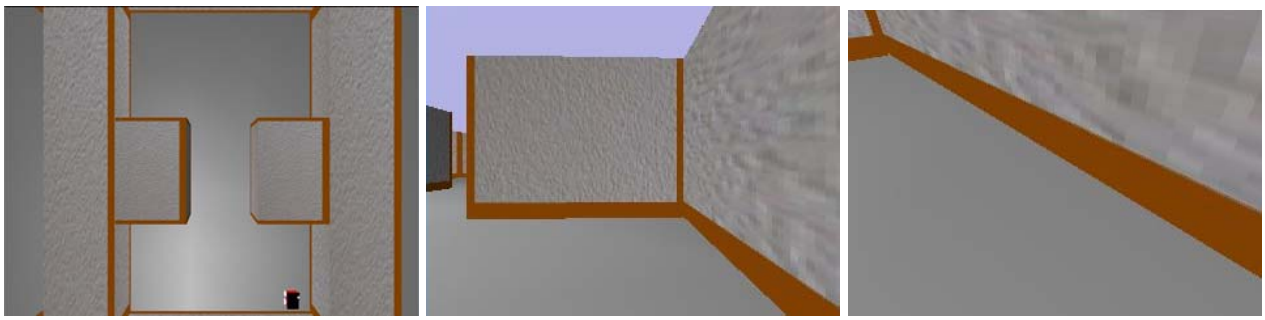

Fig. 4 Learning with the Gazebo simulator: (left) environment and mobile robot (elevated view); (center) frontal view from the robot; and (right) view from the same position with the camera inclined 35 toward the floor, and turned 40 to the right

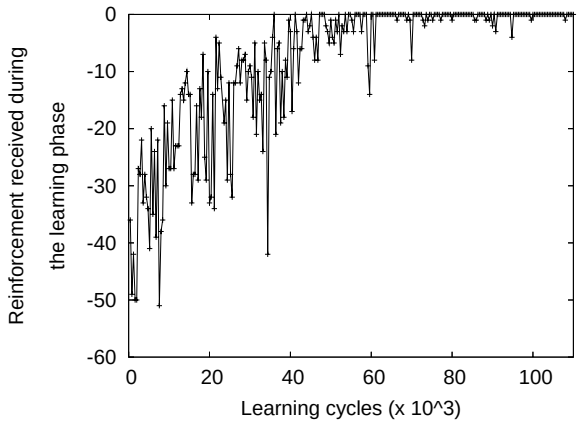

Fig. 5 Reinforcement accumulated during learning of visual contour following behaviour

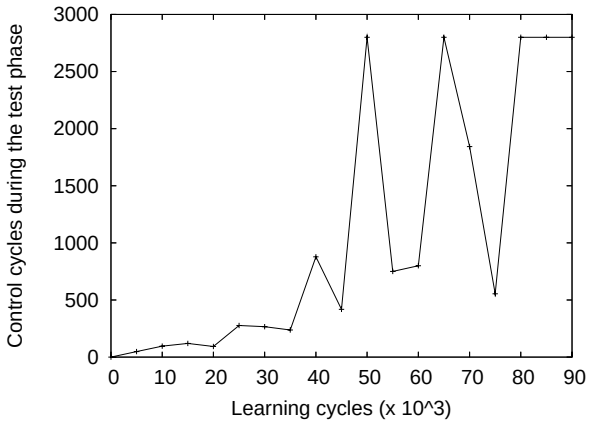

Fig. 6 Number of failure-free cycles during the test of learned contour following behaviour (maximum 2.800)

learned by the behaviour; i.e. none of them should receive negative reinforcement in at least 2,800 cycles (approximately 4 laps of our environment).

As can be seen in the diagrams, the agent learns the task in 80,000 learning cycles. Even though the instability or oscillations that occur during the learning phase are to be expected, it should be mentioned that not all states are equally "critical". For example, if the robot is very close to a wall and it carries out an incorrect action a serious error is almost inevitable. Thus, a change in the selection of that action will give rise to a high level of instability; hence the strong variations in the diagrams.

A total of 860 states were explored (overlooking those that were scarcely explored, i.e. states whose Q-values are all very low) out of a possible 6561 (see Sect. IV-A). The most common action (in 30% of states) was ”sharp turn to the right” (robot is following right contour). The remaining actions were shared out almost evenly among the states.

Figure 7 (top) shows the robot's route in the first stages of learning. The oscillations can clearly be seen. For comparison, figure 7 (bottom) shows the trajectory of the robot on attaining convergence. The set of states is created and is numbered dynamically, due to which it is difficult to make a comparison between the different learning processes. It should be mentioned that even though convergence is obtained, new states may appear even towards the end of the learning phase as the state space is enormous.

The application of the Q-learning algorithm in the implementation of a visual behaviour in robotics highlights its limitations, even though in this study it is shown that it converges. A total of 80,000 learning cycles were needed, or in other terms, 29 laps of our environment or 15 hours.

The behaviour learned is robust and functions in situations that are more complicated than that used in the learning. For example, the height of obstacles was reduced until they were no longer detected by ultrasound detectors. Nevertheless, the visual behaviour learned functioned correctly. This shows the validity and utility of using artificial vision in the implementation of behaviours in mobile robotics.

Another test carried out involved establishing gaps or divisions between obstacles (figure 8 (left)) which were not used during learning: gaps of between 20 and 40 cm were tested, and both cases gave rise to new states. Nevertheless, these were not critical and, thus, their action in not relevant and did not affect the final result. As the size of the gaps increased, more and more new states appeared, and the final result worsened.

Contour following behaviour in square obstacle (figure 8 (center)) was also tested, since open corners are the most critical situations. The systems works if an “horizon” is included into the environment in order for the upper cells (see figure 2 (right)) to be occupied, as in the training environment.

Lastly, the behaviour was tested in an extreme situation without prior training: in a passage (see figure 8 (right)). Different widths were tested: With values in excess of 2 metres (the most common in indoor environments) the system functioned adequately; when the width was reduced to 1.8 m some times the system failed; finally, when the width of the passage was reduced to 1 m the behaviour was no longer operative.

The robustness of the behaviour was also demonstrated on other tests: by distorting the acquired images with up to 30% Gaussian noise; modifying the linear velocity (constant) by around 20%; and modifying the set of learned actions (angular velocities) also by 20%.

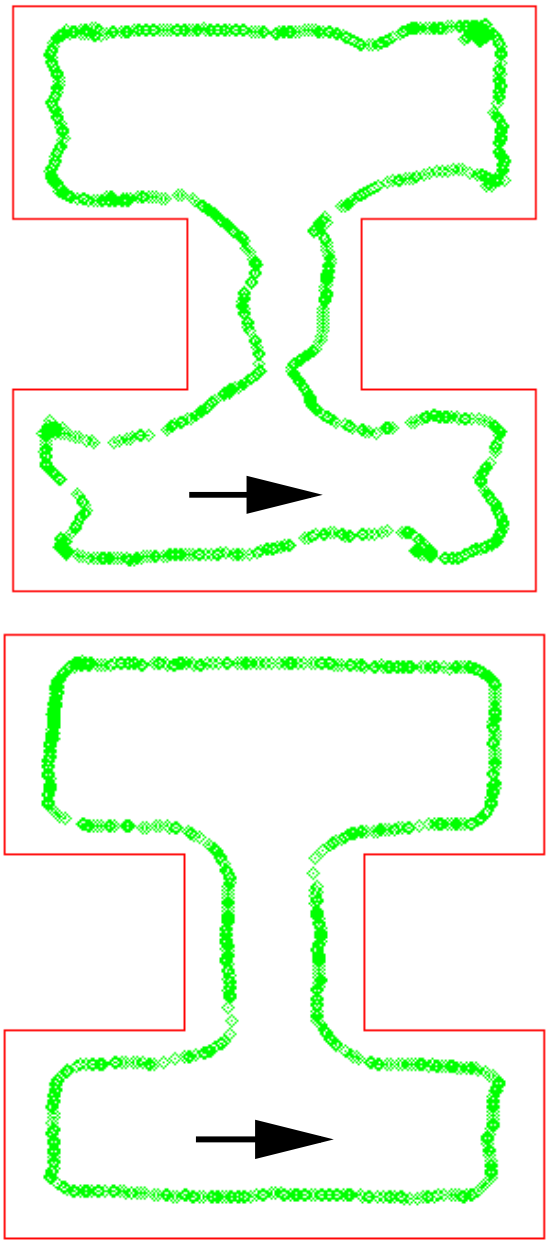

Fig. 7 Trajectories obtained during the learning of contour following behaviour: (top) initial stages; (bottom) on conclusion

VI. Conclusions and Future Work

In this work a visual contour following behaviour for the Pioneer 2 AT robot has been implemented with RL. The principal difficulty encountered is that of defining states on the basis of the image, since a delicate balance need be struck between reducing the number of states and avoiding “perceptual aliasing”. Both learning and experimentation were carried out the 3-D Gazebo simulator.

Convergence times are relatively slow (15 hours), principally due to the large number of states and low efficiency of the (*Q-learning*) RL. Nevertheless, the proposed codification and methodology is general, not specific for the task, and have proved to be efficient and valid.

Fig. 8 Examples of environments in which the robustness of the visual contour following behaviour has been demonstrated: (left) with shortened obstacles that are not detected by ultrasound sensors; (center) obstacles with gaps; (right) square obstacle; and d) in a passage

Various tests were carried out to verify the robustness of the learned behaviour. We used obstacles that were not detected by the Pioneer 2 AT's ultrasound sensors or gaps. In both cases the system generalised perfectly and the results were optimal. If the gaps were large (over 40 cm) a large number of new states appeared with respect to the training process, and the final result deteriorated. The visual contour following behaviour was also capable of negotiating an end of passage with a minimum width of two metres. As the width of the passage is reduced the behaviour worsens.

Future lines of work include carrying out tests on a real robot, using more efficient learning algorithms (e.g. $TTD(\lambda)$), applying a delayed reinforcement scheme (so that it is not necessary to "see" the contour in all images) and establishing an adaptable automatic mechanism for defining the states of RL (e.g. neural networks).

ACKNOWLEDGEMENTS

This paper was supported in part by the Xunta de Galicia and Spanish Government under Grants PGIDIT04-TIC206011PR and TIN2005-03844, respectively.

REFERENCES

[1] C.V. Regueiro, M. Rodríguez, J. Correa, D.L. Moreno, R. Iglesias, and S. Barro, "A control architecture for mobile robotics based on specialists," in *Intelligent Systems: Technology and Applications*, C. Leondes, Ed., vol. 6, pp. 337–360. CRC Press, 2002.

[2] U. Nehmzow, *Mobile Robotics: A Practical Introduction*, Springer, 2003.

[3] R. Iglesias, C.V. Regueiro, J. Correa, and S. Barro, "Implementation of a Basic Reactive Behavior in Mobile Robotics Through Artificial Neural Networks," in *Proc. oIWANN*. 1997, vol. 1240 of *LNCS*, pp. 1364–1373, Springer Verlag.

[4] T. Nakamura and M. Asada, "Motion sketch: Acquisition of visual motion guided behaviors," in *IJCAI*, 1995, pp. 126–132.

[5] M. Mucientes, R. Iglesias, C.V. Regueiro, A. Bugarín, and S. Barro, "Fuzzy temporal rule-based velocity controller for mobile robotics," *Fuzzy sets and systems*, vol. 134, pp. 83–99, 2003.

[6] D. Driankov and A. Saffiotti, Eds., *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, vol. 61 of *Studies in Fuzziness and Soft Computing*, Springer-Verlag, 2001.

[7] J.R. Millán, D. Posenato, and E. Dedieu, "Continuous-action Q-Learning," *Machine Learning*, vol. 49, pp. 247, 265, 2002.

[8] J. Wyatt, "Issues in putting reinforcement learning onto robots," in *10th Biennal Conference of the AISB*, Sheffield, UK, April 1995.

[9] D.L. Moreno, C.V. Regueiro, R. Iglesias, and S. Barro, "Making use of unelaborated advice to improve reinforcement learning: A mobile robotics approach," in *Proc. International Conference on Advances in Pattern Recognition (ICAPR)*. 2005, vol. 3686 of *LNCS*, pp. 89–98, Springer-Verlag.

[10] E. Zalama, J. Gómez, M. Paul, and J.R. Perán, "Adaptive behavior navigation of a mobile robot," *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, vol. 32, pp. 160–169, 2002.

[11] D.L. Moreno, C.V. Regueiro, R. Iglesias, and S. Barro, "Using prior knowledge to improve reinforcement learning in mobile robotics," in *Towards Autonomous Robotic Systems (TAROS)*, 2004.

[12] R. Iglesias, C.V. Regueiro, J. Correa, and S. Barro, "Supervised Reinforcement Learning: Application to a Wall Following Behaviour in a Mobile Robot," in *Proc. IEA*. 1998, vol. 1416 of *LNAI*, pp. 300–309, Springer Verlag.

[13] C. Gaskett, L. Fletcher, and A. Zelinsky, "Reinforcement learning for a vision based mobile robot," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2000, vol. 1, pp. 403–409.

[14] T. Martínez-Marín and T. Duckett, "Fast reinforcement learning for vision-guided mobile robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005, pp. 4170–4175.

[15] M.J.L. Boada, R. Barber, and M.A. Salichs, "Visual approach skill for a mobile robot using learning and fusion of simple skills," *Robotics and Autonomous Systems*, vol. 38, pp. 157–170, 2002.

[16] J.V. Ruiz, P. Montero, F. Martín, and V. Matellán, "Vision based behaviors for a legged robot," in *Proc. Workshop en Agentes Físicos (WAF)*, 2005, pp. 59–66.

[17] U. Nehmzow, "Vision processing for robot learning," *Industrial Robot*, vol. 26, no. 2, pp. 121–130, 1999.

[18] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni-directional vision for robot navigation," in *Proc. IEEE Workshop on Omnidirectional Vision*, 2000, pp. 21 – 28.

[19] R.S. Sutton and A. Barto, *Reinforcemente Learning, an introduction*, MIT Press, 1998.

[20] L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[21] D. Burschka, S. Lee, and G. Hager, "Stereo-based obstacle avoidance in indoor environments with active sensor re-calibration," in *Proc. Int. Conf. on Robotics and Automation*, 2002, vol. 2, pp. 2066–2072.

[22] G. Gini and A. Marchi, "Indoor robot navigation with single camera vision," in *Proc. Pattern Recognition in Information Systems (PRIS)*. 2002, pp. 67–76, ICEIS Press.

[23] T. Taylor, S. Geva, and W.W. Boles, "Monocular vision as a range sensor," in *Proc. CIMCA*, 2004, pp. 566 – 575.

[24] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 4, no. 2, pp. 123 – 148, 2000.

[25] J. Michels, A. Saxena, and A.Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proc. Int. Conf. on Machine Learning*, 2005.

[26] E. Delage, H. Lee, and A.Y. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006, vol. 2, pp. 2418–2428.