Situated Learning of a Behavior-Based Mobile Robot Path Planner

Yao Shu (姚 殊) and Zhang Bo (张 钹)

Department of Computer Science and Technology, Tsinghua University
Beijing 100084

Received July 30, 1993; revised December 5, 1993.

Abstract

In this paper, we propose a behavior-based path planner that can self-learn in an unknown environment. A situated learning algorithm is designed which allows the robot to learn to coordinate several concurrent behaviors and improve its performance by interacting with the environment. Behaviors are implemented using CMAC neural networks. A simulation environment is set up and some simulation experiments are carried out to rest our learning algorithm.

Keywords: Situated learning, behavior-based, path planning, CMAC neural networks.

1 Task Decomposition

We build our path planner based on Brook's subsumption architecture[1]. In Brook's approach, the overall task is decomposed into several concurrent behaviors, each behavior has its own applicability conditions specifying when it is appropriate, and a priority ordering is pre-defined to resolve conflicts among behaviors.

In path planning, the task of the robot is to approach a target while avoiding obstacles. We decompose the task into three behaviors. They are Avoid, Steer and Advance. Fig.1 illustrates the overall structure of our path planner. Avoid behavior has the highest priority. If a collision occurs or is likely to occur, the foremost task is to avoid obstacles. Steer behavior is next in priority. Whenever the forward direction of the robot deviates from the target, the robot will turn a fixed angle in the direction of the target. Advance behavior has the lowest priority. If the robot is neither in a collision situation nor in a deviation one, it should keep on advancing toward the target.

Behaviors are implemented using CMAC neural networks[2]. For the Avoid network, the inputs are sonar signals and the outputs are turning angles to avoid obstacles. A threshold T_a is defined for the Avoid network. The Avoid behavior will be triggered either when some contact sensors are activated or when the output of the network exceeds the threshold. For the Steer network, the inputs are the same as those of Avoid network and the outputs are not turning angles but activation levels. A threshold T_s is also defined for the Steer network. The Steer behavior will be triggered only when the forward direction of the robot deviates from the target and the output of the network is below the threshold.

Supported by the National Project of Key Fundamental Researches and National '863' High-Tech. Program.

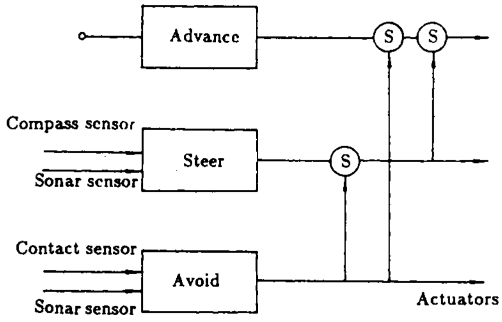

Fig.1. The overall structure of the behavior-based path planner.

2 The Simulated Environment

Fig.5 illustrates the simulation environment used in our experiment, which is a 500×500 room-like terrain populated with obstacles. The robot is represented by a 30×50 rectangle; the "nose" indicates the head of the robot. Dark shaded polygons represent obstacles and a small dark shaded disc represents a target.

The kinematics of the robot is very simple. It only has two kinds of actions. One is translation; the other is rotation.

Three kinds of sensors are mounted on our robot, which are simulated sensors modeled after the real sensors. They are sonar sensors, contact sensors and compass sensors.

3 The Situated Learning Algorithm

For each behavior a learning algorithm is designed, which will be executed when the corresponding behavior is triggered. The Avoid behavior learns whenever the Advance behavior causes a collision. At such circumstances, first, the robot will translate to a new situation according to a potential-like calculation. Second, a turning angle is decided according as which side of the robot collides with the obstacles. Third, the Avoid network learns to associate the new situation with the turning angle. Thus the learning occurs not at the colliding point but at some point nearby. With the local generalization ability of the CMAC network, the robot will turn earlier when it is likely to collide with an obstacle.

The learning of the Steer behavior is different. It depends on what happens in the next control cycle. If the Steer behavior is followed immediately by an Avoid behavior, it means that the Steer behavior in the previous circle is inappropriate and should be suppressed. The robot learns to associate the previous situation with a higher activation level. After several trials of learning the robot will learn to suppress the Steer behavior in inappropriate situations.

4 Experimental Results

4.1 Test of the Situated Learning Algorithm

The workspace depicted in Fig.5 is employed in the situation. For all the CMAC networks, the input discretization is 50, the generalization constant is 32, the learning rate is 0.3 and the physical memory size is 4003. No attempt has been made to search for the best set of parameter values. The robot is first located in the lower middle part of the environment. A trial is considered to end only when the robot reaches the target. In each trial the robot starts at the same location.

Figs.2 and 3 show the paths taken by the robot during the first and tenth trial, respectively. Fig.4 displays the number of collisions and the number of undesirable Steer behaviors. We define the undesirable Steer behavior as the one followed immediately by an Avoid behavior.

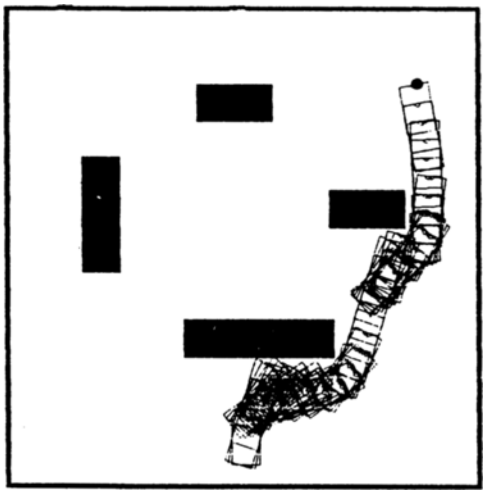

Fig.2. The path taten by the robot during the first trial.

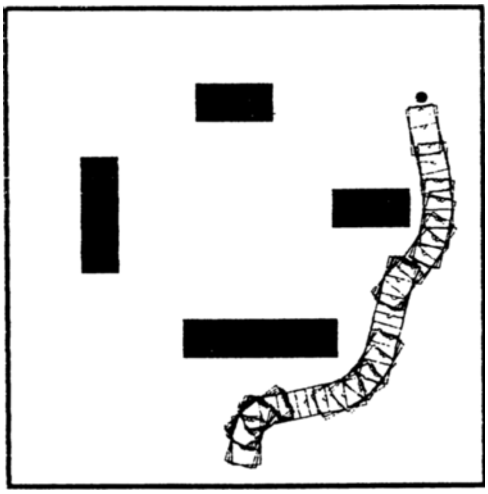

Fig.3. The path taken by the robot during the tenth trial.

Fig.4. The number of collisions and the number of undesirable Steer behaviors during each trial. Horizontal axis is the trial number. Verticalaxis is the numbers.

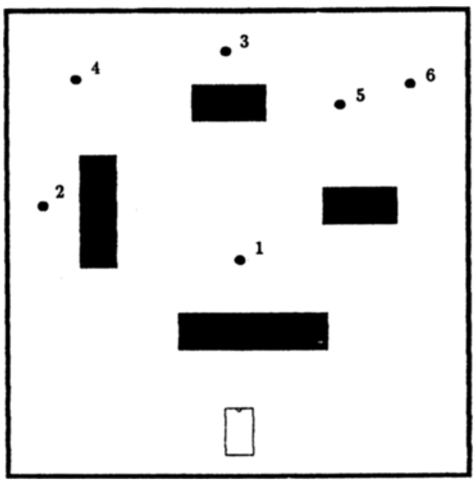

Fig.5. The learning environment with six different targets.

It can be seen from Fig.2 that during the first trial the robot displayed a clumsy zig-zag path and many collisions occurred. It is not a surprise since at beginning the robot did not know when to avoid obstacles and when not to steer toward the target. Therefore, after turning away from obstacles and advancing one step it turned back again and was led to a new collision. At this stage of performance, the robot was dominated by its basic reflexes and the sonar signals had not functioned yet. After several trials of learning, the robot learned to avoid obstacles when certain patterns of sonar signals were perceived. It also learned to suppress the Steer behavior when there were obstacles between itself and the target. As a result, in the fifth trial the number of collisions and the number of undesirable Steer behaviors were greatly decreased, which are 3 and 1 respectively. During the tenth trial no collision occurred and the Steer behavior was triggered only when the robot was away from the obstacles.

From Fig.3, we can see that the robot displayed a "follow wall" like behavior which was not preprogrammed by the designer but learned by itself by interacting with the environment. In this sense this kind of behavior is *emergent.*

4.2 Learning Interference and Generalization

In this experiment we discuss the learning interference and the generalization ability of our learning algorithm. First, we want to know whether the experience the robot learns afterwards will interfere with what it has learned. Second, we want to know how well the robot performs when it is put into an environment different from the one it encountered before.

In Fig.5, six different targets are used to train the robot. The robot is first located at the lower middle part of the environment. For each target, the robot performs 10 trials. In each trial the robot starts at the same location. The labeling number indicates the training order of the corresponding target.

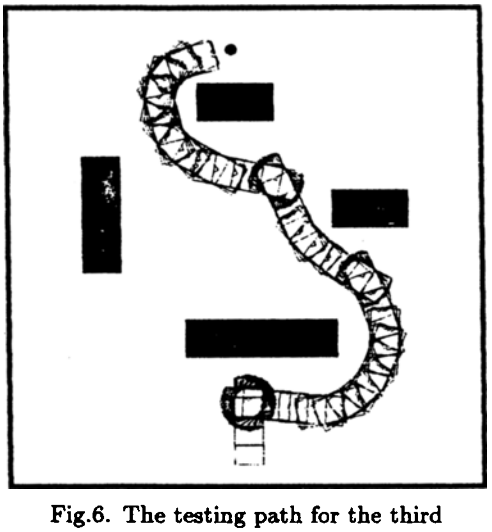

Fig.6. The testing path for the third target after the total 60 trials are over.

After the total 60 trials, targets in the previous trials are tested. In Fig.6, the path for the third target is displayed. It can be seen that the robot performs well. Therefore, we can say that what the robot learns afterwards does not interfere with what it has learned.

In Fig.7, some new obstacles are added to the environment. It can be seen that the robot still performs well. It takes a different path and there are only one collision and two undesirable Steer behaviors.

In Fig.8, the experienced robot is put into a new environment. It is again located at the lower middle part of the environment. In this case, the robot also performs well except two undesirable Steer behaviors.

These simulation results show that our learning algorithm has some generalization ability. This ability is partly due to the local generalization ability of the CMAC network. It is also due to the similarity of the environment. In the environment of Fig.5 the robot has learned by itself how to perform in a number of typical situations, such as, in front of a wall, at a turning angle, in some kinds of corridor. Thererfore, when it is put into a new environment, it can generalize from these typical situations and performs well. On the other hand, although the robot has performed 60 trials the different patterns of situation the robot has encountered are still very limited and there might exist some circumstances in which the robot performs badly. But this is not a problem for our robot because whenever the robot does not perform well, i.e. some collisions or undesirable Steer behaviors occur, the learning mechanism will be triggered and the robot will eventually learn to perform well in such new situations. In fact, it is one of the advantages of our situated learning algorithm that the robot situates in the environment and improves its performance by interacting with the environment.

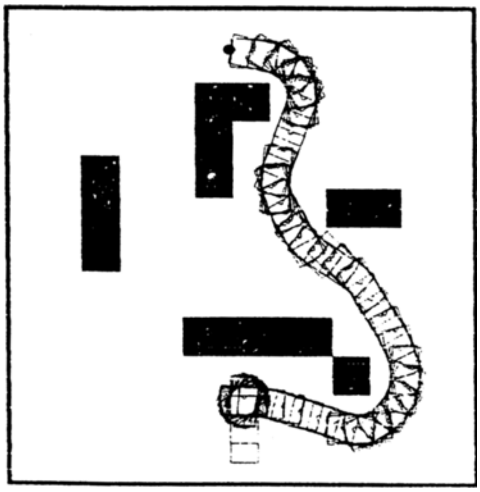

Fig.7. The testing path for the third target after some new obstacles are added into the environment.

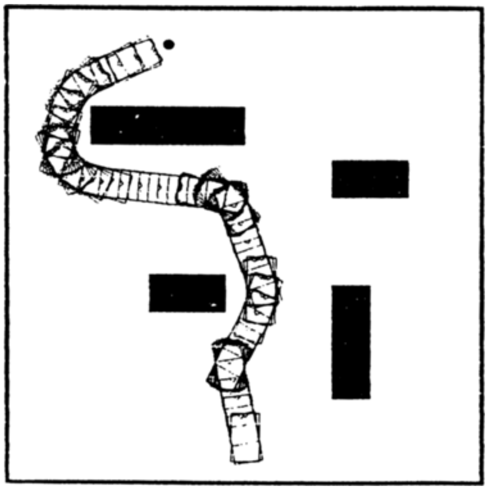

Fig.8. The path taken by the robot when it is put into a new environment.

5 Conclusions

In this paper, we present a behavior-based mobile robot path planner that can improve its performance by interacting with the environment. The path planning problem here is a nontrivial one since the environment is unknown and the robot must learn to coordinate obstacle avoiding behaviors with target approaching behaviors. However, from the simulation results shown above, we can see that after several trials of learning our robot can avoid its previous clumsy behaviors and performs well.

References

[1] Brooks R A. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 1986, RA-2: 14–23.

[2] Albus J S. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Trans. ASME, J. Dynamic Syst. Meas. Contr.*, 1975, 97: 220–227.

[3] Verschure P F *et al.* Distributed adaptive control: The self-organization of structured behavior. *Robotics and Autonomous Systems*, 1992, 9: 181–196.

[4] Mahadevan S, Connell J. Automatic programming of behavior-based robots using reinforment learning. *Artificial Intelligent*, 1992, 55: 311–365.

Yao Shu is a Ph.D. candidate of Dept. of Computer Science, Tsinghua University. His main interests include neural network, machine learning and robotics.

For the biography of **Zhang Bo** please see p.111 of this volume.