Generalization Performance of Vision Based Controllers for Mobile Robots Evolved with Genetic Programming

Renaud Barate
ENSTA - UEI, Paris, France
renaud.barate@ensta.fr

Antoine Manzanera
ENSTA - UEI, Paris, France
antoine.manzanera@ensta.fr

ABSTRACT

We present a genetic programming system to design automatically vision based obstacle avoidance algorithms adapted to the current context. We use a simulation environment to evaluate the controllers. By restricting the structure of the algorithms to facilitate the compromise between obstacle avoidance and target reaching, we improve the generalization performance of the algorithms.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*program synthesis*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*depth cues*

General Terms

Algorithms

Keywords

Vision, obstacle avoidance, robotic simulation, generalization

1. INTRODUCTION

We want to design obstacle avoidance controllers using only monocular vision. One commonly used method is based on the perceived movement and the computation of optical flow but those systems don't cope well with thin or lowly textured obstacles. Other systems use appearance information, typically texture or color, to discriminate the floor from the rest of the scene. Martin tried evolutionary techniques to parameterize this method automatically [2]. The results are generally more robust but the floor must be flat and uniform. Saxena designed a system to estimate depth from monocular images but it needs a ground truth (depth images) for the learning phase [3].

As none of those methods is robust enough to deal with all kind of environments, we want our system to automatically select the kind of visual features that allow obstacle detection in a given environment. More, it should adapt the whole obstacle avoidance controller to use those features efficiently. For that, we propose to use genetic programming as it allows us to evolve algorithms with little *a priori* on their structure and parameters. We use a simulation environment for the evaluation of the algorithms. The goal of the robot is to go to a fixed target location while avoiding obstacles. We showed that when all the algorithm is designed by the evolutionary process, the compromise between moving toward the target point and avoiding obstacles is difficult to achieve [1]. To overcome this problem, we designed a more restricted algorithmic structure that facilitates this compromise. We will show that those controllers are more generic as they still can solve this problem even when the environment changes.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

2. EVOLUTION OF VISION ALGORITHMS

We use a grammar based genetic programming system, as introduced by Whigham [4], to generate and evolve the vision algorithms. These algorithms are built with primitives that transform the input data (video image, target direction and distance) in an output command used to drive the robot. This command is composed of the requested forward and angular speeds. The grammar describes the rules that will generate, transform and parameterize these algorithms. The primitives that can be used for that are:

- Image spatial filters (Gaussian, Laplacian, Gabor, etc.)
- Image temporal filters (recursive mean for instance)
- Optical flow computation
- Optical flow projection (on an axis, vector norm, time to contact)
- Windows integral computation (operator that transforms an image into a scalar value by computing the mean pixel values on several windows in the image)
- Scalar operators (addition, subtraction, etc.)

In this paper we compare two kinds of controllers. The first ones have a free structure and all the algorithm is built by the genetic programming system (the detailed grammar used for this can be found in [1]). The second ones have a restricted structure that facilitates the compromise between obstacle avoidance and target reaching. Fig. 1 shows the global algorithm used for those structure-restricted controllers. The obstacle detection and command generation parts are still built by the genetic programming system.

We use a simulation environment for the evaluation of the different obstacle avoidance algorithms. We first record sequences where the robot is manually driven to the target location while avoiding obstacles. Those sequences contain the images from the simulated camera and the commands issued by the human at each time step. In a first phase of

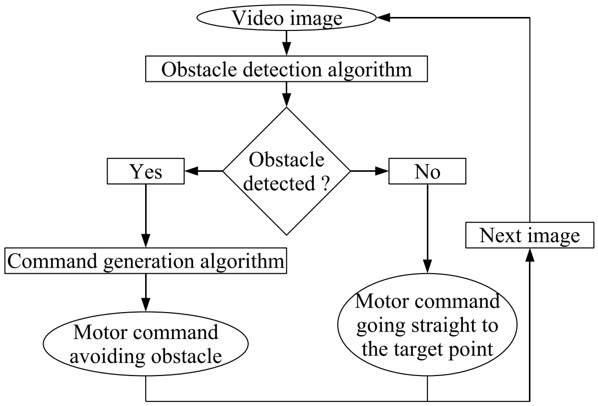

Figure 1: Overview of the global fixed structure used for structure-restricted controllers.

the evolution process, the algorithms are evaluated on their ability to match these commands throughout the sequences. In a second phase, the robot moves freely in the simulation environment during each experiment. The algorithms are evaluated on their ability to reach the target location while avoiding obstacles. Each phase lasts for 50 generations. We showed in [1] that this two-phase evolution system greatly improves the results compared to one-phase evolution.

3. EXPERIMENTS AND RESULTS

We compare the controllers created by the evolution with the two types of structure presented before. The difference appear when we compare their trajectories. The structure-restricted controller applies a simple back and forth strategy: it goes forward toward the goal as long as there is no obstacle in front of it, and it goes backward turning on itself when it faces an obstacle. Nevertheless this simple strategy is sufficient as the robot reaches the target point in each case and with only a small number of contacts. With the structure-free controller, the robot moves much faster but it doesn't make a relevant use of the target direction information, so it will never find the target point unless by chance.

In order to test the genericity of the evolved controllers, we designed a second environment with the same appearance but where the obstacles and the start and target points have been moved around. In this environment, the structure-restricted controllers generally outperform the structure-free ones. This is clearer when we consider the trajectories of the robot with the two kinds of controllers (Fig. 2). The structure-restricted controller adapts quite well to those new conditions and almost always reaches the target point, with only a few contacts with the obstacles. The structure-free controller avoids the obstacles correctly but since it does not make a relevant use of the target direction information, it doesn't reach the target location.

Those experiments show that using a customized structure for the evolution of algorithms can bring significant benefits, especially when we must find a compromise between two different behaviors. Indeed we can evolve very efficient obstacle avoidance controllers or controllers that move toward the goal with a free structure, but only by using a kind of rule-based structure we can evolve controllers that move toward a target point while avoiding obstacles.

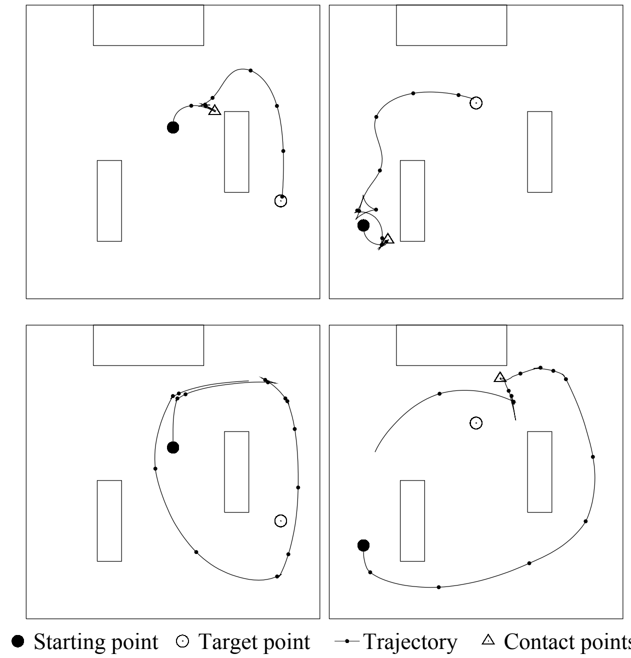

Figure 2: Top: Trajectories in the generalization environment with the structure-restricted controller. Bottom: Trajectories with the structure-free controller.

4. CONCLUSION

We presented the benefits of using a specific algorithmic structure for the evolution of vision based obstacle avoidance controllers with genetic programming. This structure facilitates the compromise between going toward a target point and avoiding obstacles, therefore creating more generic controllers. Our goal is now to adapt the system to create controllers for a real robot and to design a high-level controller able to select its behavior depending on the context in real-time.

Note: The full 8-page version of this paper can be found at http://www.ensta.fr/~barate/papers/2008-Barate-GECCO-full.pdf

5. REFERENCES

[1] R. Barate and A. Manzanera. Evolving Vision Controllers with a Two-Phase Genetic Programming System Using Imitation. *From Animals to Animats 10, Proceedings of the tenth International Conference on the Simulation of Adaptive Behavior, SAB'08*, To Appear, 2008.

[2] M. Martin. Evolving visual sonar: Depth from monocular images. *Pattern Recognition Letters*, 27(11):1174–1180, 2006.

[3] A. Saxena, S. Chung, and A. Ng. 3-D Depth Reconstruction from a Single Still Image. *International Journal of Computer Vision*, 76(1):53–69, 2008.

[4] P. Whigham. Grammatically-based genetic programming. *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 33–41, 1995.