Obstacle detection by Evolutionary Algorithm: the Fly Algorithm

Olivier Pauplin, Jean Louchet, Evelyne Lutton, Michel Parent
INRIA Rocquencourt, Projets IMARA & COMPLEX
BP 105, 78153 Le Chesnay Cedex France
olivier.pauplin@inria.fr, jean.louchet@inria.fr, evelyne.lutton@inria.fr, michel.parent@inria.fr
http://www-rocq.inria.fr/imara/, http://fractales.inria.fr/

Abstract

Artificial vision is a key element in robots autonomy. The Fly algorithm is a fast evolutionary algorithm designed for real time obstacle detection using pairs of stereo images. It aims to be used in particular in the fields of mobile robotics and automated vehicles. Based on the Parisian approach, the Fly algorithm produces a set of 3-D points which gather on the surfaces of obstacles. This paper describes the use of the Fly algorithm for obstacle detection in a real environment, and a possible use for vehicle control is presented.

Keywords: evolutionary algorithm, stereovision, vision systems for robotics, obstacle detection, automatic control

1 Evolutionary algorithms

Evolutionary algorithms [1] [2] [3] are a now widespread class of optimisation algorithms based on Darwin's theory of Evolution. That theory assumes that a population of individuals, characterised by their genes, evolves toward a better adaptation to its environment according to laws of natural selection. The more adapted an individual is, the more likely he is to stay alive and to transmit his genes and characteristics to his offspring. Genes mutations may occur and maintain diversity in the population.

Evolutionary algorithms manipulate individuals evaluated by a function, called fitness function, in a way similar to biological Evolution. The general diagram of such algorithms is presented in figure 1.

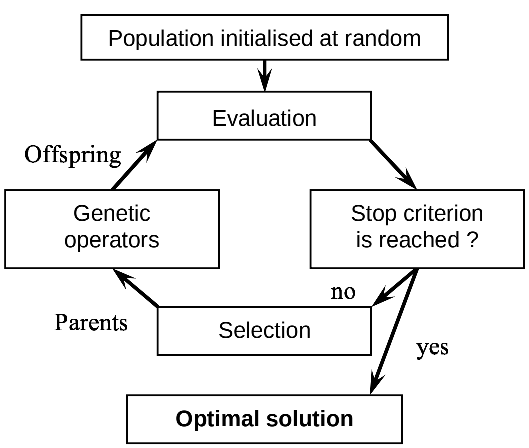

Figure 1: General layout of genetic algorithms

Where:

- the population is a group of individuals;
- an individual is defined by his genes $X = (x_1, x_2, \ldots, x_n)$, usually coordinates in the search space;
- evaluation is the calculation of each individual's fitness value;
- selection eliminates part of the population, keeping preferably the best individuals;
- evolution applies genetic operators (crossover, mutations…), leading to new individuals in the population.

2 The Fly algorithm

Parisian evolution [4] is an evolutionary optimisation approach for which each individual of the population is part of the solution, and the global solution to a problem is given by a population of individuals. The Fly algorithm [5] [6] is a special case of Parisian evolution for which individuals (the "flies") are defined as 3-D points with coordinates (x, y, z). The aim of the algorithm is to drive the whole population - or a significant part of it - into suitable areas of the search space, corresponding to the surfaces of visible objects in the scene.

The population of flies is initialised at random inside the intersection of two cameras' field of view. Flies then evolve following the steps of evolutionary algorithms.

2.1 Evaluation

The fitness function used to evaluate a fly compares its projections on the left and right images given by the cameras. If the fly is on an object's surface, the projections will have similar neighbourhoods on both images and hence this fly will be attributed a high fitness.

Figures 2 and 3 illustrate that principle. Figure 3 shows neighbourhoods of two flies on left and right images. On that example, Fly1, being on an object's surface, will be given a better fitness than Fly2.

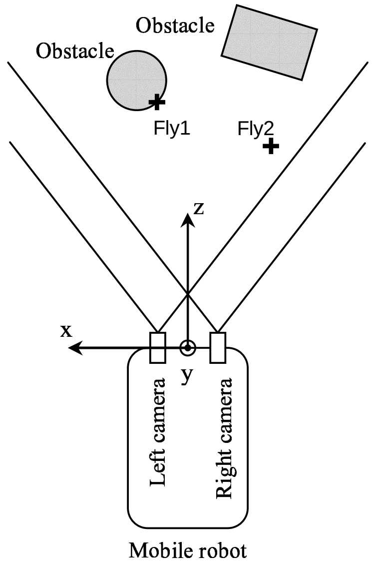

Figure 2: Example of device using the Fly algorithm, showing two flies from the population (top view)

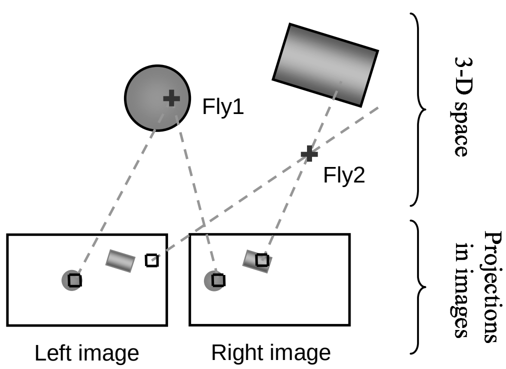

Figure 3: Projections of two flies in left and right images

The mathematical expression of the fitness function is [7] [8]:

$$F = \frac{|\nabla(M_L)|.|\nabla(M_R)|}{\sum_{colours} \sum_{(i,j)\in N} [L(x_L+i, y_L+j) - R(x_R+i, y_R+j)]^2}$$

where:
- (x_L, y_L) and (x_R, y_R) are the coordinates of the left and right projections of the current individual;
- $L(x_L+i\ ,\ y_L+j)$ is the grey value at the left image at pixel $(x_L+i\ ,\ y_L+j)$, similarly with R for the right image;
- N is a neighbourhood introduced to obtain a more discriminating comparison of the fly's projections;
- $|\nabla(M_L)|$ and $|\nabla(M_R)|$ are Sobel gradient norms on left and right projections of the fly. That is intended to penalise flies which project onto uniform regions, i.e. less significant flies.

2.2 Selection

Selection is elitist and deterministic. It ranks flies according to their fitness values and retains the best individuals (around 40%).

A sharing operator [7] [8] reduces the fitness of flies packed together and forces them to explore other areas of the search space.

2.3 Genetic operators

The following operators are applied to selected individuals.

- Barycentric cross-over: given two parents F_1 and F_2, the algorithm builds their offspring F such as:

$$\vec{OF} = \lambda \vec{OF_1} + (1-\lambda)\vec{OF_2}$$

with λ chosen at random in the interval [0,1].

- Gaussian mutation: adds a Gaussian noise to each one of the three coordinates of the mutated fly. The mutation rate is set to 40%, parisian algorithms normally using a higher mutation rate than conventional evolutionary algorithms.

- Another operator, "immigration", is used to improve exploration of the search space, creating new individuals at random. It ensures a constant exploration of the search space, whose high-fitness regions evolve as the scene in front of the cameras changes.

3 Robot simulator

The original way the scene is described by the population of flies led to adapt classical robot navigation methods in order to use the results of the Fly algorithm as input data. Boumaza [7] [9] developed a simulator of a robot moving in a simplified environment, to test theoretically control methods using the output of the Fly algorithm.

The simulator showed the possibility to build guidance methods based on the output of the Fly algorithm. Our current work consists in transferring and extending these control methods to real life situations.

Figure 4: Application example of the Fly algorithm

4 Real life experiments

Figure 4 shows an application example of the Fly algorithm. Flies (black dots) concentrate on obstacles and on regions where the grey level gradient is high, for example on the roadsides. The numerator of the fitness function prevents flies from getting trapped into uniform regions (sky, road surface, etc.).

The three coordinates of each fly being known, the population of flies gives a rough description of the real 3-D scene.

4.1 Control

With the intention to use the Fly algorithm in the field of automatic driving - or at least assisted driving, we developed a strategy to make the program quantify the probability that an obstacle is in front of the vehicle. The aim is to deliver a slow down or stop order when an obstacle appears close enough in the field of vision, in order to avoid frontal collision.

The general idea to achieve this goal is to see each fly as the source of a “warning value”, as high as:

- the fly is near the vehicle;
- the fly is in front of the vehicle (i.e. close to the z axis);
- the fly has a good fitness.

Beforehand, flies useless for this specific application have there fitness value penalised, and thus have high probability to be eliminated by the algorithm’s mechanisms. We considered such flies are:

- flies with a height above 2 metres;
- flies with a height under 10 centimetres (detecting the ground);
- flies at a distance of more than 16 metres from the vehicle.

An experimental analysis led us to choose the simple following formula for the warning value of a fly:

$$warning(fly) = \frac{F}{x^2 \times z}$$

where F is the fitness value of the fly, and z and x its coordinates as shown on figure 2.

For $|x| < 0.5$ metre we consider $x = 0.5$ metre, and for $z < 1$ metre we consider $z = 1$ metre. That is made to avoid giving excessive warning values to flies with a not necessarily good fitness but with a very small x or z coordinate. Moreover, obstacles within a range of half a metre to the left or to the right from the centre of the vehicle ($|x| < 0.5$ metre) are equally dangerous, and are logically treated in the same way.

The warning function was built in order to give high warning values to flies for which the three coefficients F, $1/x^2$ and $1/z$ are simultaneously high. Indeed a fly with a low fitness value (thus probably not on an obstacle), far from the vehicle or not in front of it, doesn’t show evidence of an imminent collision. Experiments with a $1/x$ factor instead of $1/x^2$ didn’t give satisfying results, as it tended to overestimate the importance of flies not in front of the cameras.

4.2 Results

To validate the algorithm, we tested it on two stereo pairs of images representing a pedestrian crossing the street in front of the vehicle. In the first case (figures 5 and 6) the pedestrian is further from the cameras than in the second case (figures 7 and 8). Figure 5 does not show a case of emergency breaking, whereas figure 7 shows a situation closer to a collision.

The landscape is not easy to process, with trees and textures ground which can “trap” flies away from real obstacles.

Results are obtained using two commercial CCD cameras and a computer (Pentium 2GHz). The population of flies is 5000. One generation takes about 10 milliseconds. The update of the population and the calculation of the warning values are done in a quasi-continuous way, and the system needs about 10 to 30 generations to react to a new event in the scene.

Figures 5 and 7 show the 250 best flies of the resulting population. Flies appear as black crosses. We note that, on both figures 5 and 7, part of the population of flies gathered on the pedestrian.

Figures 6 and 8 show the same (x,y) view as figures 5 and 7, with only flies represented. Flies appear as spots as dark as their warning value is high.

We note the algorithm delivers higher warning values in figure 8 than in figure 6.

Figure 5: Pedestrian at 8 metres from the cameras, at the roadside

Figure 6: Warning values of figure 5 flies

Figure 7: Pedestrian at 4 metres from the cameras, on the middle of the road

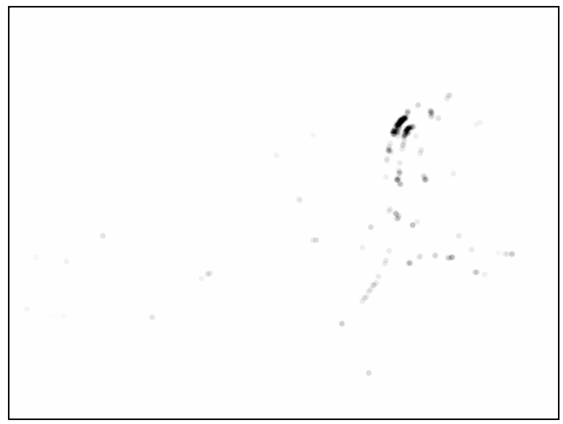

Figure 8: Warning values of figure 7 flies

A global warning value can be defined as the mean of the warning values of a population. In the first case, this mean is 0.40, whereas in the second case it is 0.85. The high difference between those two values confirms that they can be used as a discriminating element between the two situations.

5 Conclusion

The Fly algorithm has proved a valid method for obstacle detection in outdoor environments. The simplicity of the fitness function used opens the way to real time applications. Real time vehicle control based on the information of flies (coordinates, fitness value) has been developed.

Our future work will be directed toward developing guidance algorithms for mobile robots in real life situations, and to integrate them into a vehicle of IMARA project.

6 Acknowledgements

We thank Amine Boumaza for his important contribution to the development of the code used in our experiments.

This research was funded in part by the IST Programme of the European Commission in the CyberCars project (http://www.cybercars.org/).

7 References

[1] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA (1989)

[2] I. Rechenberg, "Evolution strategy", J.M. Zurada, R.J. Marks II, C.J. Robinson, (Eds.), *Computationnal Intelligence Imitating Life,* IEEE Press, Piscataway, NJ, pp. 147-159 (1994)

[3] J.-P. Rennard, *Vie artificielle*, Vuibert, ISBN 2-7117-8694-3, pp 241-242 (2002)

[4] P. Collet, E. Lutton, F. Raynal, M. Schoenauer, "Individual GP: an alternative viewpoint for the resolution of complex problems", *Genetic and Evolutionary Computation Conference GECCO99*, Morgan Kauffmann, San Francisco, CA (1999)

[5] J. Louchet, "From Hough to Darwin: an individual evolutionary strategy applied to artificial vision", *Artificial Evolution, European Conference, AE 99,* Dunkerque, France, Selected papers, Springer Verlag, Lecture Notes in Computer Science 1829 (1999)

[6] J. Louchet, "Stereo analysis using individual evolution strategy", *Internat. Conf. on Pattern Recognition, ICPR2000,* Barcelona, Spain (2000)

[7] A. Boumaza, J. Louchet, "Dynamic Flies: Using Real-Time Parisian Evolution in Robotics", *EVOIASP 2001*, Lake Como, Italy (2001)

[8] J. Louchet, M. Guyon, M.-J. Lesot, A. Boumaza, "Dynamic Fies: a new pattern recognition tool applied to stereo sequence processing", *Pattern Recognition Letters*, No. 23 pp. 335-345 (2002)

[9] A. Boumaza, "Introduction de techniques d'évolution artificielle en vision tridimensionnelle et en robotique mobile", *Thèse Université René Descartes*, Paris (2004)