Stereo Analysis Using Individual Evolution Strategy

Jean Louchet
Ecole Nationale Supérieure de Techniques Avancées
32 boulevard Victor, 75739 PARIS cedex15, France
louchet@ensta.fr

Abstract

This paper presents an individual evolutionary strategy devised for image analysis applications. The example problem chosen is obstacle detection using a pair of cameras. The algorithm evolves a population of three-dimensional points ('flies') in the cameras fields of view, using a low complexity fitness function giving highest values to flies likely to be on the surfaces of 3-D obstacles. The algorithm uses classical sharing, mutation and crossover operators. The final result is a fraction of the population rather than a single individual. Some test results are presented and potential extensions to real-time image sequence processing and mobile robotics are discussed.

1. Introduction

The goal of this paper is to present one application of individual evolution strategies to stereo image analysis.

Artificial evolution is a powerful optimisation tool [4][6] that has already been successful in several computer vision applications [8][12]. Its principle is to use Darwinian evolution's principles to evolve a population of potential solutions in order to optimise a given fitness function. However, applications in image processing suffer from the fact that solutions of computer vision problems are often complex scene descriptions and it requires heavy calculations to manipulate populations of them. Moreover, one can object to the general philosophy of conventional evolutionary algorithms, that it is spillage to evolve a large population during generations and finally only retain one individual and discard all the rest. An answer to this is the individual approach [1] which consists in representing the solution as the whole population or a large fraction of it, rather than as a single individual. Thus, the solution of the problem is split into several simpler primitives, each of them represented as one individual in the population.

This concept appears to be well adapted to computer vision, as complex scene descriptions may often be easily split into several independent, simpler primitives. The algorithm presented in this paper uses one of the simplest possible primitives, 3-D points[1]. The idea is to evolve a population of 3-D points in the space of the scene, using a fitness function such that the 3-D points ("flies") concentrate upon the objects to be detected in the scene.

In stereovision, the fitness function may be defined as follows. If the fly is located on the apparent surface of an object, then in the two images, the pixels corresponding to the calculated flies' projections will probably have the same grey levels[2]. Conversely, if the fly is not on the surface of an object, thanks to the non-uniformity of objects and illumination, its projections will have no reason to get the same grey level. The algorithm presented expresses this property into a fitness function in order to evolve the flies.

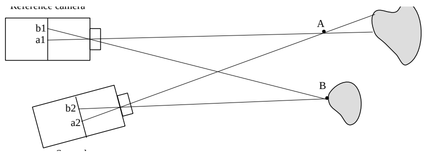

Fig. 1 : Pixels $b1$ and $b2$, projections of fly B, have identical grey levels. Pixels $a1$ and $a2$, projections of fly A, do not necessarily have identical grey levels as they correspond to two different points on the surface of the object.

The fitness function evaluates the degree of similarity of the pixel surroundings of the projections of the fly onto each image: this ensures highest fitness values for the individuals lying on the surface of an object.

2. Evolving Flies

2.1. Geometry and Fitness Function

The two cameras used will be called the reference and the second camera[3]. The fly's projections coordinates are (x_R, y_R) in the reference camera image and (x_S, y_S) in the second one. The calibration parameters are supposed to be known and allow to calculate x_R, y_R, x_S, y_S in function of x, y, z using classical Projective Geometry formulas [5].

The cost function can be defined as the correlation coefficient of the neighbourhoods of the fly's projections in the two images. Alternatively, it can be defined as the inverse of a mean square error function:

$$fitness(indiv) = \frac{1}{\sum_{(i,j) \in neighbourhood} (R(x_R + i, y_R + j) - S(x_S + i, y_S + j))^2}$$

where:

- $R(x_R + i, y_R + j)$ is the grey level of the reference image at pixel $(x_R + i, y_R + j)$
- $S(x_S + i, y_S + j)$ is the grey level of the second image at pixel $(x_S + i, y_S + j)$
- N is a small neighbourhood, in order to measure the match quality over several pixels.

As shown in Fig. 1, such fitness functions would give an undesirable high fitness value to any fly located in front of a uniform object, whatever its distance. In order to overcome this problem, the fitness function has to include a normalizing factor to provide an acceptable trade-off between giving high fitness values to non significant pixels and giving undue advantage to highly contrasted ones. In practice, this factor only needs to be calculated using the grey levels from one of the two images. Additionally, the fitness function is slightly altered in order to reduce its sensitivity to lower spatial frequencies, through subtracting a local mean to the images.

Thus, most pixel-level calculations are contained in the fitness function. Lets us now examine the operators of the evolutionary resolution engine.

2.2. Artificial Evolution

The *initial population* is generated in the vision cone of the reference camera, truncated using an arbitrary clipping distance. An individual's *chromosome* is the triple (x, y, z) which contains the individual's coordinates in the reference camera's coordinate system, where Oz is the camera axis. The statistical distribution of the individuals is chosen in order to obtain a uniform distribution of their projections in the reference image. In addition, we choose a uniform distribution of the values of z^{-1} such that the individuals stay beyond an arbitrary clipping line (minimum distance): this implies that the individuals' probability density is lower at high distances. The geometrical calibration parameters of the cameras are supposed to be known. This allows, for each individual (x, y, z), to calculate its image coordinates in each camera and calculate its fitness value.

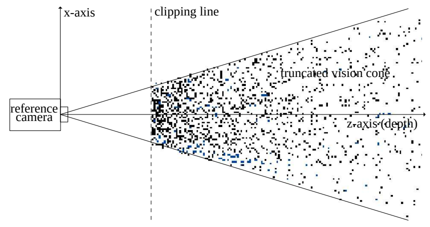

Fig. 2 : The fly population is initialised inside the field of vision of the reference camera.

3. Evolutionary Operators

Selection is deterministic. It uses a ranking process based on the individuals' fitness values. It has not been optimised and is the most time consuming operation in the algorithm.

Mutation is the main operator which allows an extensive exploration of the search space. It uses a quasi-Gaussian noise added to the individuals' chromosome parameters X, Y and Z, with a fixed standard deviation.

In order to prevent the population from getting concentrated into a very small number of maxima, 2-D *sharing* allows to reduce the fitness values of individuals located in crowded areas. Thus, the presence of one individual A with coordinates (x, y, z) lowers the fitness values of all the individuals whose projection on the reference image is close enough to the projection of individual A. Sharing uses two control parameters: sharing radius and sharing coefficient.

Many real-world images contain convex primitives as straight lines or planar surfaces. We translated this into a *barycentric crossover operator* which builds an offspring randomly located on the line between its parents: the offspring of two individuals with space coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) is the individual whose space coordinates (x_3, y_3, z_3) are defined by :

$$x_3 = \lambda x_1 + \mu x_2 \ ; \ y_3 = \lambda y_1 + \mu y_2 \ ; \ z_3 = \lambda z_1 + \mu z_2$$

where the weights $\lambda, \mu \ (\lambda + \mu = 1)$ are chosen using a uniform random law in the $[0,1]$ interval[4].

Best results are usually obtained with around 40% mutation and 20% crossover rates, but this depends on the test images chosen and on the population size (smaller populations require higher crossover rates).

4. Parameter Sensitivity and Convergence Results

4.1. Results on synthetic images

We used the synthetic "Money" test images[5] which correspond to a simple scene and allows easier result readability.

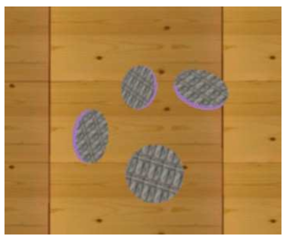

Fig. 3 : "Money image", left

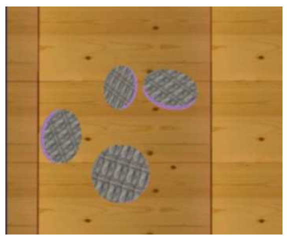

Fig. 4 : "Money image", right.

Convergence results using a 5000 individual population, a 50% mutation rate and a 10% crossover rate at different stages of convergence are shown on Fig. 5. The scene is seen from above (vertical projection of the population), showing the axes x_l (horizontal) and z (vertical). Effects of mutation and crossover rates are shown on Fig. 6.

Fig. 5 : Results after 10 (top left), 50 (top right), 100 (bottom left), 1000 (bottom right) generations.

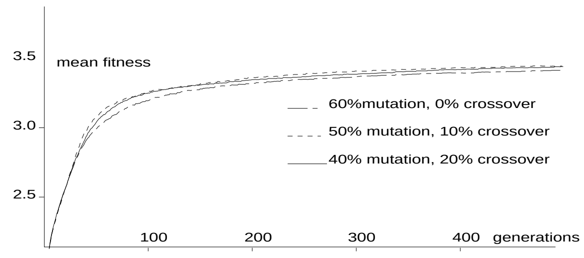

Fig. 6 : Evolution of average fitness values for a 5000 individual population using three different combinations of mutation and crossover rates.

The best average fitness is obtained here with 50% mutation and 10% crossover rates. The bottom curve corresponds to a mutation-only evolution. CPU times (based on a 366MHz Linux i686 PC, ANSI C, gcc) are about 1.75 seconds for initialising plus 28 milliseconds per generation with a population of 5000 (independently of image type and size).

4.2. Example Result on Real Images

The 760×560 images on Figs. 7 and 8 were taken using a parallel set of two monochrome cameras. Parameters are: 5000 individuals, 100 generations, 40% mutation, 20% crossover, sharing radius 2, sharing coeff. 0.3.

Fig. 7: left image

Fig. 8 : right image

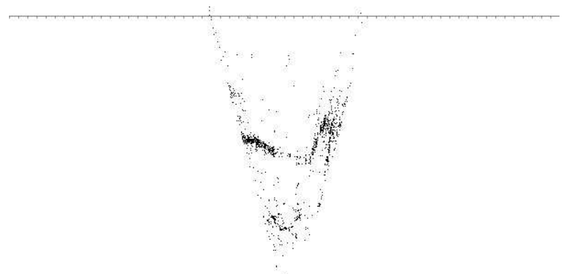

Fig. 9 : plan view

On Fig. 9, one can see the two cabinet sides, the beginning of the wall on the right and the front half-circle of the stool.

5. Image Sequences and Motion Analysis

It is often considered that genetic algorithms and evolution strategies are slow and therefore not well adapted to real-time applications. However :

- Speed is not the only issue with real-time applications. Real-time means the ability to exploit incoming data and react to them as fast as needed by the end user. Evolution Strategies are generally able to adaptation, i.e. to cope with modifications of the fitness function during the algorithm's execution [13], unlike most other optimisation methods.
- The processing speed of an evolutionary Strategy is strongly dependent on the computational complexity of the fitness function - which is fairly simple in our case.

The algorithm is being extended to stereo image sequences (Fig. 10) in order to be implemented into a mobile robot vision system [3]. In order to be able to process objects' apparent movements more efficiently, the fly's larger chromosome now includes its speed vector, used to update the fly's position at the next frame. Flies with appropriate speed vectors get a selective advantage, which allows the flies' velocities to converge towards the velocities of the objects in the scene, relative to the robot's coordinate system. Mutation and crossover are applied both to positions and speeds.

Fig. 10 : results on one stereo pair (left) and a sequence of 4 stereo pairs (right) - robot speed is unknown.

6. Conclusion

Unlike conventional approaches to stereovision, the progressive stereo analysis method described in this paper does not require any image pre-processing or segmentation. The gradually increasing accuracy of results and the integration of velocities into the flies' chromosomes should be of interest in Robotics applications, to process image sequences containing moving objects.

While the Hough transform [7] uses a vote technique in order to explore a parameter space, here the parameter space is explored by the evolving population, where each individual tests a pixel property based on a model of image formation. There is no obvious general rule telling which approach is more efficient, but it appears that even with the static flies' only three parameters, a complete Hough-style implementation of the 3-dimensional parameter space would already result in much higher complexity. The benefits of the individual evolutionary approach are:

- high speed processing (partly due to the non-exhaustive search),
- progressive accumulation of knowledge making it possible to use the results at any stage of the algorithm,
- real-time compliance, as the fitness function may include time-dependent parameters, e.g. like odometric or other sensor-based data.

Future research includes egomotion estimation and obstacle avoidance directly using fly data .

References

[1] Pierre Collet, Evelyne Lutton, Frederic Raynal, Marc Schoenauer, "Individual GP: an Alternative Viewpoint for the Resolution of Complex Problems", *GECCO99*, Orlando, Florida, July 1999.

[2] R. C. Eberhart, J. Kennedy, "A new Optimizer Using Particles Swarm Theory", *Proc. 6th Int. Symposium on Micro Machine and Human Science*, Nagoya (Japan), IEEE service Centre, Piscataway, NJ, 39-43, 1995

[3] D. B. Gennery, *Modelling the Environment of an Exploring Vehicle by means of Stereo Vision*, Ph.D. thesis, Stanford University, June 1980.

[4] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[5] R. M. Haralick, "Using Perspective Transformations in Scene Analysis", *CGIP 13*, 1980, 191-221.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, Univ. of Michigan Press, 1975

[7] P. V. C. Hough, *Method and Means of Recognizing Complex Patterns*, U.S. Patent n°3, 069 654, December 1962.

[8] Evelyne Lutton, Patrice Martinez, "A Genetic Algorithm for the Detection of 3D Geometric Primitives in Images", *12th ICPR*, Jerusalem, Israel, October 9-13, 1994 / INRIA technical report # 2210.

[9] David Marr, *Vision*, W. H. Freeman and Co., San Francisco, 1982.

[10] I. Rechenberg, "Evolution Strategy", in J.M. Zurada, R.J. MarksII, C.J. Robinson, *Computational Intelligence imitating life*, IEEE Press (1994) 147-159.

[11] John O'Rourke, "Motion Detection using Hough technique", *IEEE conference on Pattern Recognition and Image Processing*, Dallas 1981, 82-87.

[12] G. Roth and M. D. Levine, "Geometric Primitive Extraction using a Genetic Algorithm", *IEEE CVPR Conference*, 1992, 640-644.

[13] Ralf Salomon and Peter Eggenberger, "Adaptation on the Evolutionary Time Scale: a Working Hypothesis and Basic Experiments", *Third European Conference on Artificial Evolution*, Nîmes, France, October 1997, Springer Lecture Notes on Computer Science no. 1363, 251-262.

[1] The particle swarm approach [2] also uses 3-D points as primitives but operators are kinematic rather than genetic (selection, mutation, crossover).

[2] This is essentially true with Lambertian (matt) surfaces where rediffusion of incident light is isotropic. Most usual non glossy surfaces differ slightly from the Lambertian model, but this may be at least partly taken into account in the fitness function (see below). Reflections on glossy surfaces may give rise to virtual objects and wrong 3-D interpretation, independently of the class of image processing algorithm being used.

[3] In the standard Stereovision configuration, they are the left and right cameras.

[4] It is generally accepted that such a crossover operator has contractive properties which may be avoided by using a larger interval. However the idea in our application is that contrasts are often higher on objects' edges and therefore higher fitness values and higher individuals densities are likely to be obtained on objects' edges. The goal of the crossover operator is to fill in surfaces whose contours are easier to detect, rather than to extend them. This is confirmed by experiments showing that there is no benefit in using a wider interval.

[5] "Money" image pair, ©INRIA - Mirages project.