The Fly Algorithm for Robot Navigation

Rodrigo Montúfar-Chaveznava[1], Mónica Pérez-Meza[2], and Ivette Caldelas[3]

[1] Engineering Division, ITESM Santa Fe, Av. Carlos Lazo 100, Del. A. Obregón,
01389 México D.F., Mexico
`rmontufar@itesm.mx`
[2] Universidad de la Sierra Sur, Miahuatlán de Porfirio Díaz, Oaxaca, Mexico
[3] Instituto de Investigaciones Biomédicas, Universidad Nacional Autónoma de México,
México, D.F., Mexico

Abstract. The fly algorithm is a strategy employed for 3D reconstruction of scenes, which employs the genetic algorithms and stereovision principles to determine clusters of points corresponding to different objects present in scene. The obtained reconstruction is partial, but enough to recognize obstacles in the robot space work. This 3D reconstruction strategy also allows to know the dimensions of the detected objects. Many parameters are involved in the fly algorithm, and then it is difficult to assign the optimal values for the best performance. In this work we test different parameters values, analyze the results and present some improvements to the algorithm considering the fly algorithm can be employed in robot navigation.

Keywords: The fly algorithm, Evolutive Strategies, Vision stereo, Robotics.

1 Introduction

In particular, in this work we are looking for a strategy to reconstruct the world where an autonomous mobile robot operates using a stereovision system. The reconstruction of the entire world is practically unmanageable, but, a partial reconstruction is possible and it provides the basic idea of the world structure. In this sense, the use of both, stereovision theory and evolutive algorithms, allow us to reconstruct partially the world. Note that some parameters are involved in the performance of the reconstruction, principally those related to the evolutive algorithm. In this work we analyze these parameters to determine the optima, considering that external factors such as illumination can modify the performance of the reconstruction strategy.

2 Reconstruction 3D

Usually, the 3D reconstruction algorithms have been limited to the schemes defined by Marr [1], which are based on the image exploitation and extraction of all information necessary for a scene interpretation. These schemes have three stages: segmentation, reconstruction and recognition. At first stage, the low level primitives are extracted from images. Next, these primitives are merged to produce a 2D reconstruction. Finally, the object identification is performed by comparison.

Many reconstruction algorithms have emerged from these approximations, and some of them employ stereovision techniques to find correspondences between images and triangulation processes to obtain a 3D structure.

A. Gottscheber et al. (Eds.): EUROBOT 2008, CCIS 33, pp. 119–127, 2009.

Stereoscopy is a very popular technique to carry out 3D reconstruction, based on the process performed by the biological vision. It requires at least a pair of images, captured from different positions in a scene, to obtain the structure of the space or infer the 3D position of spatial points. Stereoscopy is an active area of computer vision research [2, 3] based on the works achieved in the field of photography at the beginning of the last century. Vision stereo problem is easily solved by triangulation if the camera calibration parameters are known. The determination of camera calibration parameters is also another problem in 3D reconstruction that has been exhaustively studied and several solutions have been proposed [4, 5, 6].

3 Evolutionary Algorithms

Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. These algorithms work on populations of individuals or solutions, performing a parallel search of the best ones.

3.1 The Fly Algorithm

The use of evolutive algorithms for exploration in a parameters space in image analysis problems has been exhaustively explored in last years [7]. There are some works [8, 9] that extract 2D primitives using evolutive algorithms. The fly algorithm [10] is an approximation that searches in the parameters space for the best 3D model consistent with the stereo images. The scene model is defined as a cluster of 3D points or flies (Fig.1a) and the algorithm explores the 3D space looking for the most representative cluster in scene.

An optimization method is the Parisian evolution [11], which considers the global solution is given by a population and each individual is part of it. A special case of the Parisian evolution is the Fly algorithm where individuals (flies) are defined as 3D points with coordinates (x, y, z). The objective of the algorithm is to guide an important part of the population to suitable areas of the search space, corresponding to the surfaces of visible objects in scene.

The fly algorithm can be considered an image processing technique based on the evolution of a population of flies projected on a pair of stereo images. The evolution is regulated by a function of adjustment designed to make the flies converge on the objects located in the scene. This adjustment function is called the *fitness function*.

The algorithm projects the flies on the stereo images, producing a new pair of coordinates: (x_R, y_R) and (x_L, y_L), corresponding to the position of the flies in the right and left images respectively. Initially, the population of flies is created randomly in the space formed by the intersection of the view of both images. Flies then evolve following the rules of evolutionary algorithms. We identify three components in the algorithm:

The fitness function. Compares the fly projections on the left and right images. If the fly is posed on an object, the projections will have similar pixel neighborhoods on both images, producing a high fitness value. If the fly is "flying", the projections will have different pixel neighborhoods, and the fitness value will be low. The idea is illustrated in Fig. 1b. There, fly_1, which is settled on an object, has a better fitness value than fly_2. The fitness function employed is:

$$F = \frac{\left|\nabla(M_L)\right| . \left|\nabla(M_R)\right|}{\left(\sum_{colors} \sum_{(i,j)\in N} (L(x_L + i, y_L + j) - R(x_R + i, y_R + j))^2\right)} \tag{1}$$

(x_L, y_L) and (x_R, y_R) are the coordinates of the projected fly on the left and right images; $L(x_L + i, y_L + j)$ and $R(x_R + i, y_R + j)$ are the pixel color values; N is the neighborhood population introduced to obtain a more discriminating comparison of the fly projections; $\left|\nabla(M_L)\right|$ and $\left|\nabla(M_R)\right|$ are the norms of the gradients of Sobel on the projections of the fly. That is intended to penalize flies when they are posed on uniform regions.

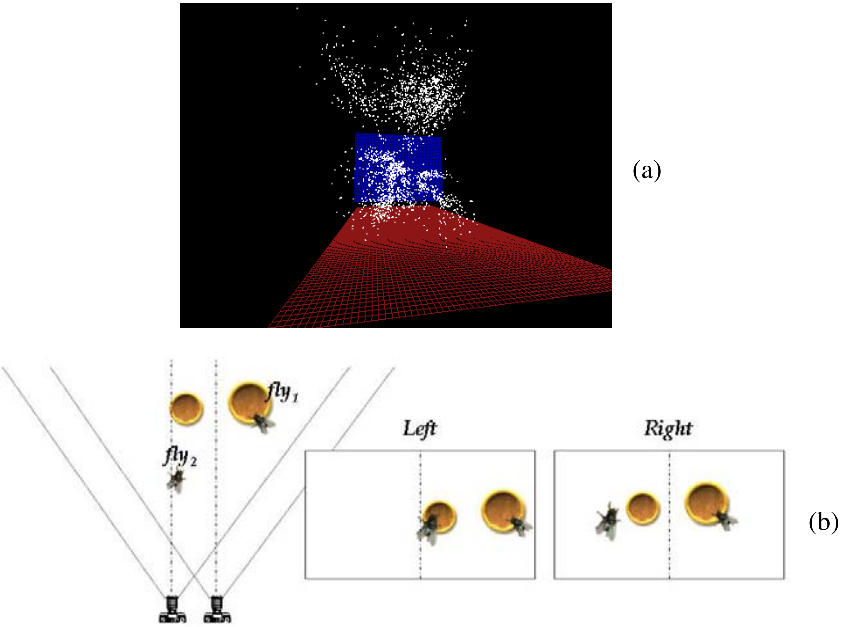

Fig. 1. (a) The space of the fly algorithm. The scene model is represented as cloud of 3D points. (b) The projection of flies: The pixel neighborhoods for fly_1 are very in stereo images; the pixel neighborhoods for fly_2 are different. This observation is employed in the fitness function.

Selection. Classifies the flies according the fitness values, preserving the best individuals. It uses sharing rules that splits the population, forcing a part of it to explore other areas on the search space by replacement. The flies with low fitness values are replaced and new flies are generated using genetic operators.

Genetic operators. We use the following operators to generate new flies:

- Barycentric cross-over. From two parents F_1 and F_2, the heir F is crated such as $\overrightarrow{OF} = \lambda\overrightarrow{OF_1} + (1-\lambda)\overrightarrow{OF_2}$, where λ is a random value between [0, 1].

- Gaussian mutation. The mutation operation adds Gaussian noise to each component of the 3D coordinate in a discriminated fly to produce a new one..
- Immigration. This operation extends the exploration area in search space, creating new individuals randomly; to ensure a constant exploration of the whole space of search.

4 Results

The performance of the fly algorithm depends on the dynamics of the world and the values provided to the involved parameters such as stop criterion, search area definition (image division), percentage of crossing, mutation and immigration, etc.

Table 1. Results obtained using different genetic operators values: Sets A, B and C

Population size	**Stop criterion**	**Set A**			
		No. iterations		*Time (ms)*	
		2 reg.	*4 reg.*	*2 reg.*	*4 reg.*
500	90 %	37	26	1096	787
	95 %	37	30	1096	906
1500	90 %	26	21	1674	1368
	95 %	26	22	1693	1449
3000	90 %	25	20	2896	2294
	95 %	26	22	2996	2490
5000	90 %	27	23	4743	4180
	95 %	27	24	4783	4409
Population size	**Stop criterion**	**Set B**			
		No. iterations		*Time (ms)*	
		2 reg.	*4 reg.*	*2 reg.*	*4 reg.*
500	90 %	27	25	813	768
	95 %	28	28	818	834
1500	90 %	22	19	1427	1259
	95 %	23	22	1455	1412
3000	90 %	22	20	2584	2309
	95 %	23	22	2674	2566
5000	90 %	26	24	4796	4262
	95 %	27	25	4868	4578
Population size	**Stop criterion**	**Set C**			
		No. iterations		*Time (ms)*	
		2 reg.	*4 reg.*	*2 reg.*	*4 reg.*
500	90 %	21	21	653	665
	95 %	23	23	709	696
1500	90 %	19	18	1231	1193
	95 %	20	19	1293	1249
3000	90 %	21	19	2424	2124
	95 %	21	20	2437	2336
5000	90 %	21	21	3793	3818
	95 %	22	22	4048	3827

4.1 Pair of Stereo Images

The pair of stereo images are divided in two and four regions. Initially, the population is equally divided and posed in each region. During execution, the standard deviation of each region is computed and employed to determine the region characteristics.

We test three sets of genetic operator percentages:

Set A: Selection 40%, Crossing 10%, Mutation 40%, Immigration 10%.
Set B: Selection 50%, Crossing 20%, Mutation 20%, Immigration 10%.
Set C: Selection 40%, Crossing 20%, Mutation 20%, Immigration 20%.

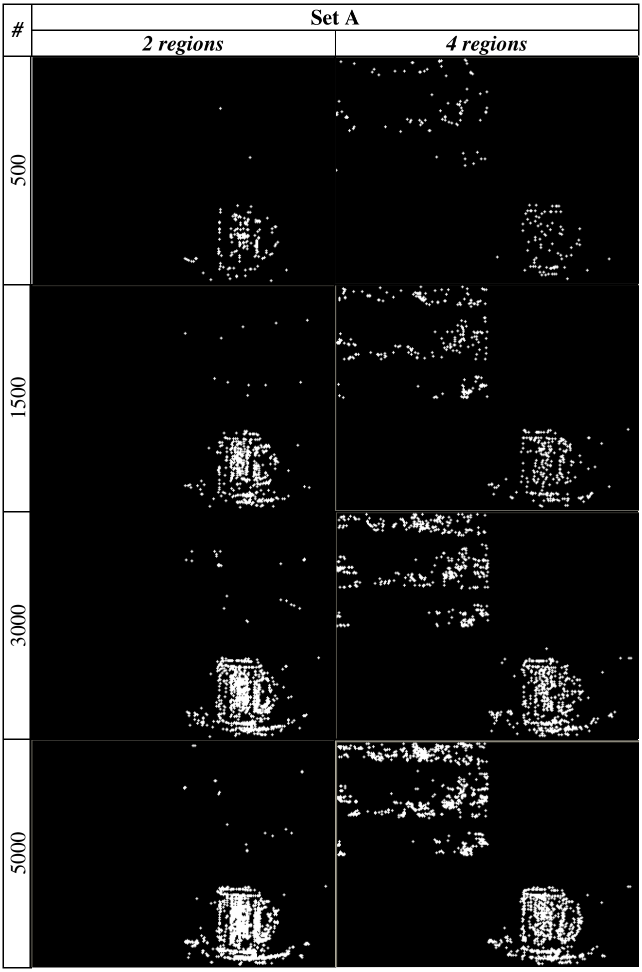

Fig. 2. Results obtained using the genetic operators set A, image division for 2 and 4 regions and population size of 500, 1500, 3000 and 5000

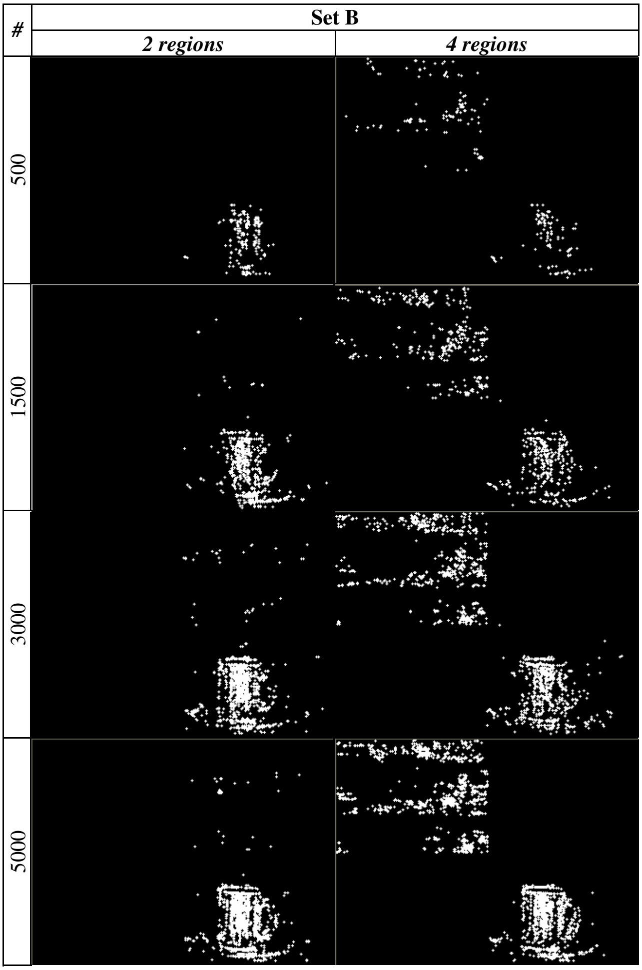

Fig. 3. Results obtained using the genetic operators set B, image division for 2 and 4 regions and population size of 500, 1500, 3000 and 5000

A threshold value was defined as a stop criterion considering the values observed in the fitness function.. We try different population sizes. The results corresponding to Sets A, B and C are presented in Table 1. We compare the number of iterations and processing time to reach the stop criterion. Results are showed in Figs. 2, 3 and 4.

4.2 Image Sequence: Robot Navigation

The system was tested in an educational robot, three paths were defined and the navigation was controlled to capture the sequence of stereo images necessary for the reconstruction. The robot has monocular vision (one camera system), and then it is necessary to provide an adequate control to get accurate camera positions. The genetic

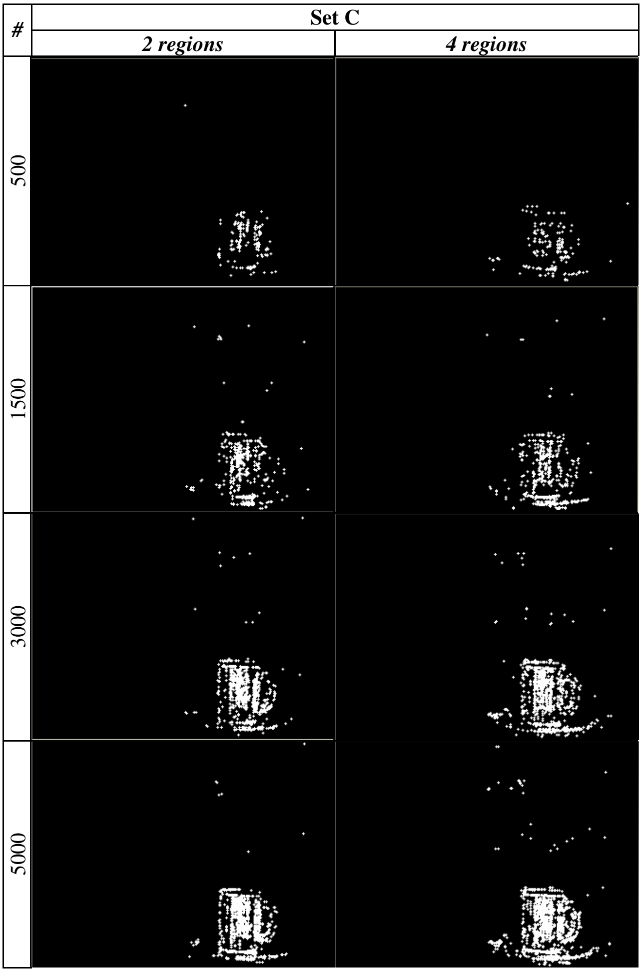

Fig. 4. Results obtained using the genetic operators set C, image division for 2 and 4 regions and population size of 500, 1500, 3000 and 5000

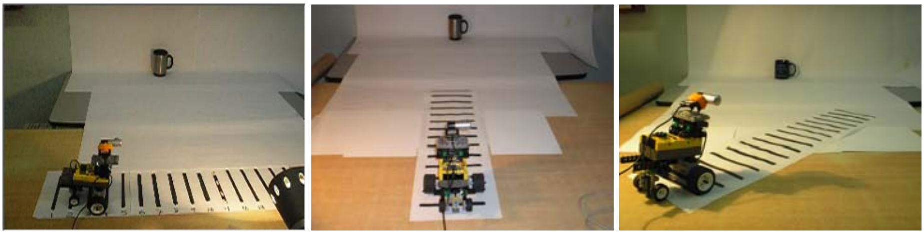

Fig. 5. Navigation paths. At each mark the robot stops and captures an image. Every pair of captured images is employed in the fly algorithm and the reconstruction carried out.

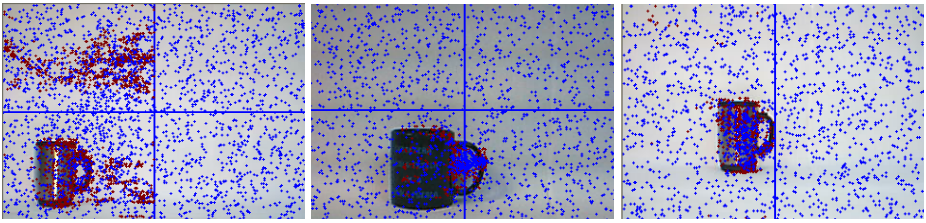

Fig. 6. Results obtained at the end of each path. The flies are posed on the object.

Fig. 7. Visual reconstructed planes *XY*, *XZ* and *YZ* corresponding to path (a). The object detected is well defined. Corresponding 3D reconstruction is showed in Fig. 1.

operators employed correspond to set A in Table 1 using 5000 flies. Fig. 5 shows the navigation paths followed by the robot. Fig. 6 shows the visual results for every path. Fig. 7 shows the different reconstruction planes obtained in the first path.

5 Conclusions

We have presented the results of different tests carried out varying the values of the genetic operators, the population size and the image division. We can say that a population of 3000 flies, the use of the set C and an image division in four regions present the best results.

A reasonable well-defined 3D reconstruction can be performed during robot navigation in harmony with the robot control. The use of additional image operators allows obtaining well defined clusters of flies and a better recognition of objects. The computed world metrics are very exact when the camera calibration is well performed and the robot navigation system provides the correct coordinates. We conclude the fly algorithm for 3D reconstruction in robot navigation is an excellent option, producing almost real-time results even with a monocular vision system.

References

1. Marr, D., Poggio, T.: A computational theory of human stereo vision. Proc. of the Royal Soc. of London. Series B, Biological Sciences 204(1156), 301–328 (1979)
2. Gutierrez, S., Marroquin, J.L.: Disparity estimation and reconstruction in stereo vision. Technical communication No I-03-07/7-04-2003. CC/CIMAT, Mexico (2003)

3. Quam, L., Hannah, M.J.: Stanford automated photogrammetry research. Technical Report AIM-254, Stanford AI Lab (1974)
4. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)
5. Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In: Proc. of the Symp. on Close-Range Photogrammetry, Falls Church, VA, pp. 1–18 (1971)
6. Tsai, R.Y.: A versatile camera calibration technique for 3D machine vision. IEEE Journal of Robotics and Automation RA-3(4), 323–344 (1987)
7. Alander, J.T.: Indexed bibliography of genetic algorithms in optics and image processing. Rep. 94-1-OPTICS, Univ. of Vaasa, Dep. of Inf. Tech. and Prod. Ec. (1995)
8. Lutton, E., Martinez, P.: A genetic algorithm for the detection of d geometric primitives in images. In: Proceedings of the International Conference on Pattern Recognition, ICPR 1994, Los Alamitos, CA, pp. 526–528 (1994)
9. Roth, G., Levine, M.D.: Geometric primitive extraction using a genetic algorithm. IEEE Trans. on Pattern Analysis and Machine Intelligence 16(9), 901–905 (1994)
10. Louchet, J.: Using an individual evolution strategy for stereovision. Genetic Programming and Evolvable Machines 2(2), 101–109 (2001)
11. Collet, P., Lutton, E., Raynal, F., Schoenauer, M.: Individual GP: an alternative viewpoint for the resolution of complex problems. In: Proc. of the Genetic and Evolutionary Computation Conference GECCO 1999, San Francisco, CA, vol. 2, pp. 978–981 (1999)