Objects Detection and Tracking using Evolutionary Algorithms

Mónica Pérez-Meza[1], Rodrigo Montúfar-Chaveznava[2], and Alberto Jaramillo-Núñez[3]

[1] Universidad de la Sierra Sur, Guillermo Rojas Mijangos s/n, Ciudad Universitaria, 70800 Miahuatlán de Porfirio Díaz, Oaxaca, Mexico
[2] Instituto de Ciencia y Tecnología, República de Chile 6, Centro, 06010, Del. Cuauhtémoc, D.F., Mexico
[3] Intituto Nacional de Astrofísica, Óptica y Electrónica, Apartado Postal 51 y 216, Puebla, 72000, Mexico
mperez@hotmail.com, rodrigo_montufar@hotmail.com, ajaramil@inaoep.mx

Abstract. The aptitude to detect objects is a present and predominant characteristic in the vision systems of the live beings. This characteristic is so important that it has been a topic of investigation in the area of artificial vision. In this paper we link 3D scenes reconstruction to evolutionary algorithms through the stereo vision that usually needs of two images captured by a pair of cameras, in order to determine the distance of the objects in relation to a reference system. From this is obtained the third dimension of the objects. In this work we propose to use only a camera, which is displaced along a path, capturing images every certain distance. As we cannot perform all computations required for the total scene reconstruction, we employ an evolutionary algorithm to partially reconstruct the scene and obtain its representation. The algorithm employed is the fly algorithm, which employ spatial points named "flies" to reconstruct the most important characteristics of the world. Flies (points in the space) are evaluated by a fitness function; the flies with better fitness are posed on the object to be detected. Finally, the center of mass of the flies with the highest fitness is obtained and a geometric form is recuperated using the parameters of the centroid, this form will be use to track the object.

Keywords: 3D Reconstruction, Artificial Vision, Stereo Vision, Evolutionary Algorithms.

1 Introduction

In this work we present a method to carry out the objects detection and tracking of a dynamic scene using stereoscopy, evolutionary algorithms and monocular vision.

Some algorithms are suitable to give solution to the stereo vision problem such as the evolutionary algorithms. In this case, we employ the fly algorithm with some modifications according to the conditions, tasks and proposed objectives.

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 11-21

Stereo vision is employed to obtain the depth of the objects present in a scene. The capture of slightly displaced images of a scene can be carried out by one of the following procedures:

Static capture. Aligning two or more cameras separated among them (similar to mammalian vision system), every camera captures one image.

Dynamic capture. Displacing one camera along a path and taking images in different positions of the path.

There are works where two or more cameras are used to obtain the reconstruction of a scene as in [4,9]. As was mentioned above, in this work we propose to use one camera and a dynamic capture. After the stereo images are obtained, we performed the 3D scene reconstruction applying projective geometry and using the fly algorithm to keep the best points in a scene of the object to be detected. These points are flies posed on the object, and with these points we obtain the centroid for tracking.

2 Stereo Vision

Stereo vision is a method for obtaining the three-dimensional view of the objects in a scene. It is determined by the distance of objects in relation with a reference system.

In stereo vision, we need two or more cameras slightly displaced one of another, for obtaining the images. The pair of images has many characteristics in common, but they also have certain differences, and these differences are called disparity [10].

3 Evolutionary Algorithms

Evolutionary algorithms manipulate individuals which are evaluated by a fitness function, in analogy to the biological evolution. The general schema of evolutionary algorithms is shown in Fig. 1.

The principal characteristics are:

- The population is a group of individuals.
- An individual is defined by genes $X = (x_1, x_2\ldots\ x_n)^T$, which particularly represents its position (X, Y, Z) in the space.
- The evaluation is the computation of the fitness value in every individual.
- The selection eliminates part of the population, keeping the best individuals.
- The evolution applies genetic operators (crossover, mutation, etc.), leading to generate new individuals in the population.
- Some kinds of evolutionary algorithms are:
- Genetic algorithms, which are a technique of programming that imitate the biological evolution as a strategy to solve problems.
- Evolutionary strategies, which are rules that define the behavior of the individual under certain circumstances.
- Genetic programming, which are specific instructions in a programming language.

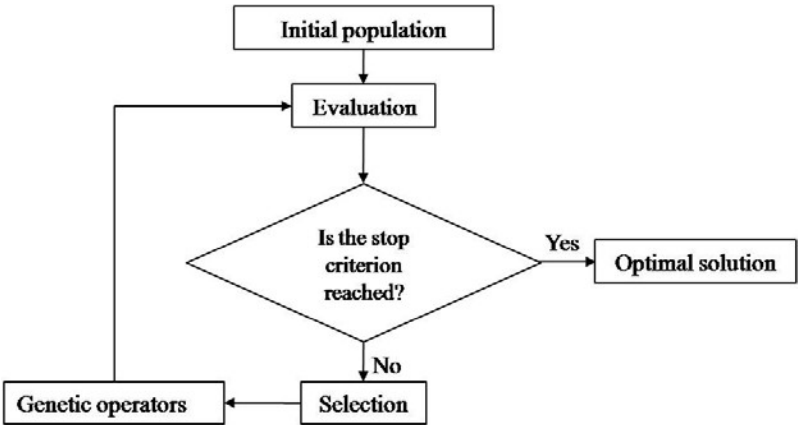

Fig. 1. Schematic description of the Genetic Algorithms method.

In this work we employ genetic algorithms. These algorithms evolve a population of individuals submitting it to random actions, like in the biological evolution, and to a selection process according to certain criterion, where the most adapted individuals are selected to survive the process and the less adapted are ruled out.

3.1 The Fly Algorithm

The fly algorithm is considered an image processing technique based on the evolution of a population of flies (points in the space) projected over stereo images. The evolution is regulated by a fitness function determined in such a way that the flies converge on the surface of an object located in the scene.

A fly is defined as a 3D point with coordinates (x, y, z). The flies are projected over a couple of despaired images by stereoscopy, producing a pair of 2D coordinates: (x_R, y_R) and (x_L, y_L) for the right and left images, respectively.

Initially, the population of flies is generated randomly in the intersection area of the view of both images and equally distributed on a certain number of regions to disperse all of them.

The fly algorithm considers the following functions:

The fitness function. The fitness function evaluates a fly, comparing projections on the images. If a fly is located on an object, the projections will have similar pixel neighborhoods on both images and the fly will have a high fitness value. This idea is illustrated in Figs. 2 and 3. Figure 3 shows the neighborhoods of two flies on left and right images. In this example, fly 1, which is located on an object, has a better fitness value than fly 2. The fitness function F defined in [9,11] is:

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

$$G = |\nabla(M_L)|, |\nabla(M_R)| \quad (2)$$

where:

- (xL, yL) and (xR, yR) are the pixel coordinates of the fly projected on the left and right images, respectively.
- L(xL + i, yL + j) and R(xR + i, yR + j) are the color values of the pixels in a neighborhood at left and right images, respectively.
- N is the dimension of the neighborhood population introduced to obtain a more discriminating comparison of the fly projections.
- $|\nabla(M_L)|$ and $|\nabla(M_R)|$ are the norms of the gradients of Sobel on the left and right projections of the fly. That is intended to penalize flies when they are located on uniform regions.
- In color images, the difference of squares in (2) on each color channel is computed. In grayscale images is computed for one channel.

Selection. Selection is elitist and deterministic. It classifies flies according their fitness values and keeps the best individuals. A sharing operator [4, 9] reduces the fitness of clustered flies packed and forces them to explore other areas on the world.

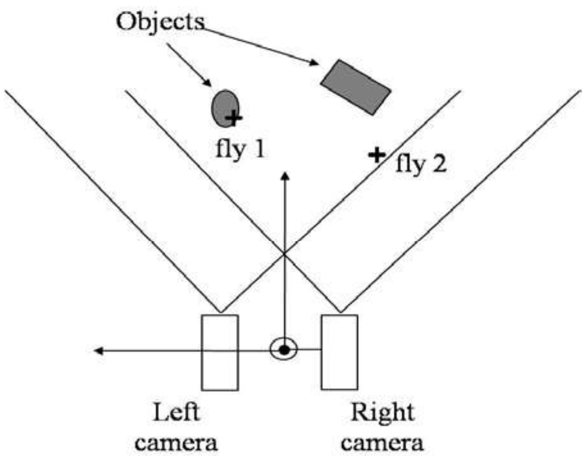

Fig. 2. View of two flies in the scene. The flies are projected on left and right image plans.

Genetic operators. We apply a pair of genetic operators to selected individuals.
Barycentric cross-over. A new individual is generated from two parents F_1 and F_2, this individual is positioned between them, as follow:

$$\overrightarrow{F_3} = \lambda\overrightarrow{F_1} + (1 - \lambda)\overrightarrow{F_2} \quad (3)$$

where λ is a random value uniformly distributed between [0, 1].
Gaussian mutation. A new fly is generated adding Gaussian noise to each coordinate of the parent fly.

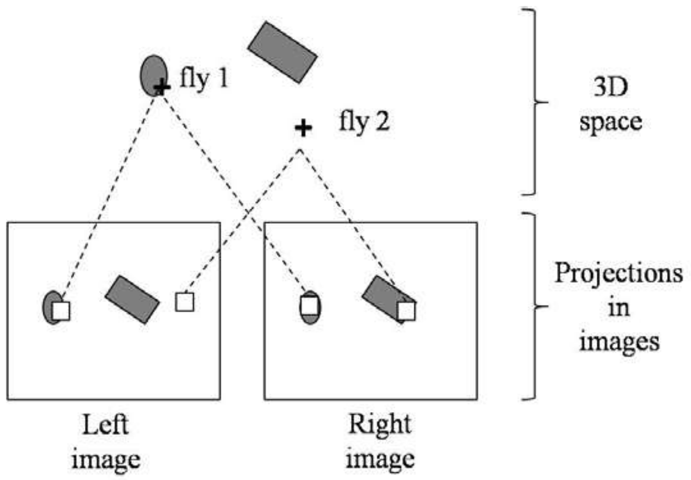

Fig. 3. Flies projections on the left and right images.

4 Image Processing

We have incorporated an image processing operator to enhance the performance of the fly algorithm and the visualization of the results. The process used was the edge detection.

4.1 Edge Detection

Edge detection is used to locate points where a sharp intensity variation is presented. The basic solution for many edge detection algorithms is the computation of local differential operators. We employ the Sobel operator [10] in the fitness function.

The Sobel operator measures the 2D spatial gradient on an image, emphasizing the regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in a grayscale image.

The Sobel operator consists of a pair of 3×3 convolution kernels. One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator [10].

5 Objects Detection and Tracking

5.1 Objects Detection

Objects recognition in an image labels sections of the image, given certain characteristics. This recognition is separated into two stages which are the detection and classification of the object.

To objects detect, we perform the edge and contours detection of the image. Each contour is considered an object to classify. For the classification, we transform the 2D representation of the contour to a radial representation that can be taken to a dimension. This representation will be made starting from the "center of mass" of the contour.

For this work, the center of mass is obtained using a proportion of flies with a high fitness value.

5.2 Object Tracking

The object tracking is calculated with the position and shape of the object in each image of a sequence of images. We can determine the motion of an object by the position and shape over a period of time.

To track a moving object through a set of images it is necessary to use the points resulting from object detection in images and to calculate the proximity of the points of the object in subsequent images.

The representation that is used in objects tracking can be from a point on the object to the 3D representation, through forms or appearances. Some forms are: points, geometric shapes, contours or shapes, articulated forms, skeletons, and so on. The representation we use in this work in objects tracking is a point which is located in the center of mass of the object.

6 Detection and Tracking System

The developed system for detection and tracking is showed in Fig. 4. It has four basic modules: Obtaining images, Reconstruction, Object detection and Object tracking. The Reconstruction module, shown in Fig. 5, has five sub-modules: Fitness function, Best flies, Crossing, Mutation and Correspondence.

The modules are:

a) Obtaining images. It obtains images (stereo pair), slightly five centimeters displaced one from the other, and these images are used in the Reconstruction module.
b) Object detection. It detects the object using a group of high fitness flies, the flies are located on the object.

c) Object tracking. After the object has been detected, its centroid is obtained with this module using the flies of high fitness.
d) Reconstruction. It performs the 3D reconstruction from the sequence of images. It has the following sub-modules.
   i) Fitness function. The fitness function is applied to the population of flies. If the color differs between each fly and its neighborhood are low, almost zero, the value of the fitness function is high for this fly and vice versa.
   ii) Best flies. In this module we obtain the flies with the highest fitness.
   iii) Crossing. This module generates new flies from two parents using the crossing function.
   iv) Mutation. In this module new flies are obtained by adding Gaussian noise to parent flies.
   v) Correspondence. In this module we obtain the correspondence of the flies projected on the right and left images by stereoscopy.

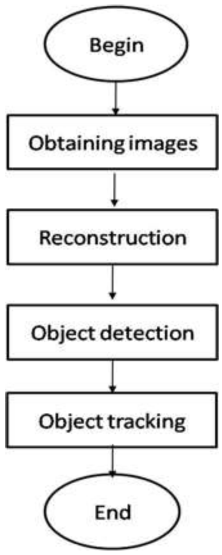

Fig. 4. Flow diagram of objects detection and tracking system.

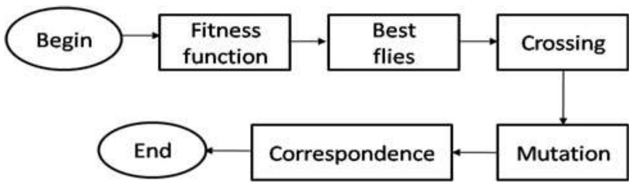

Fig. 5. Flow diagram of Reconstruction module.

Figure 5 shows the flow diagram of the Reconstruction module, where the fitness function has been computed for all individuals. We keep the best flies and the worst flies are eliminated when the operators of mutation and crossing are applied. These operators are applied at different percentages to the previous population to be projected on the right and left images. We obtain the centroid for detection and tracking with the flies with high fitness, these flies are located on the object.

7 Results

We present the results obtained using a population of 3000 flies, a set of images displaced two inches one from another on a horizontal axis. The left and right images are refreshed every 1.5 seconds approximately. In this exercise we obtain the reconstruction in the following way: the right image substitutes the left image and a new image is placed in the location corresponding to the right image. Next, with the best flies we obtain the centroid and so, we perform the object tracking. This process is showed in Fig. 6. The parameters employed in the fly algorithm are:

- Flies preserved every generation: 35%,
- Flies generated by crossing: 5%,
- Flies generated by mutation: 50%,
- Flies randomly generated: 10%.

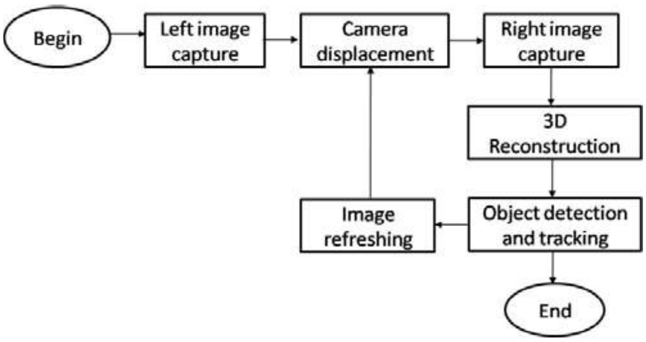

Fig. 6. Schematic description of the objects reconstruction, detection and tracking.

Figure 7 shows the left and right images. The initial population of flies is also presented. After 30 iterations, the best flies are in red, located on the object and the rest of the population is in blue. The centroid is obtained to perform object detection and tracking, as presented in Fig. 8. Next, the camera is translated two inches on the horizontal axis, then the left image is discarded, the right image takes left image place and a new image is placed in the location corresponding to the right image. This new disposition is shown in Fig. 9. We can observe that neither the flies nor the centroid are placed on the object.

Figures 10 to 12 show the results of the second, third and fourth pairs of images after 59, 89 and 117 executions. The object is detected and tracked; the flies are posed on the objects again.

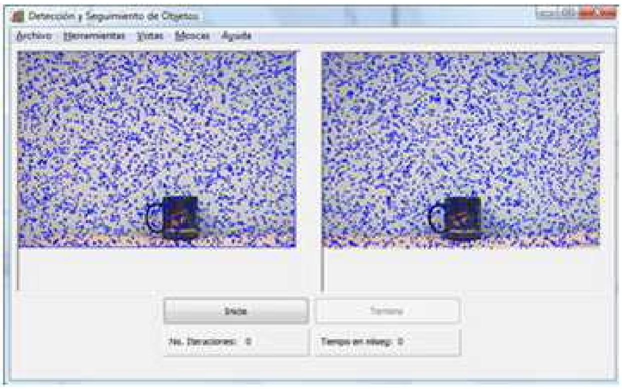

Fig. 7. Left and right images with initial population.

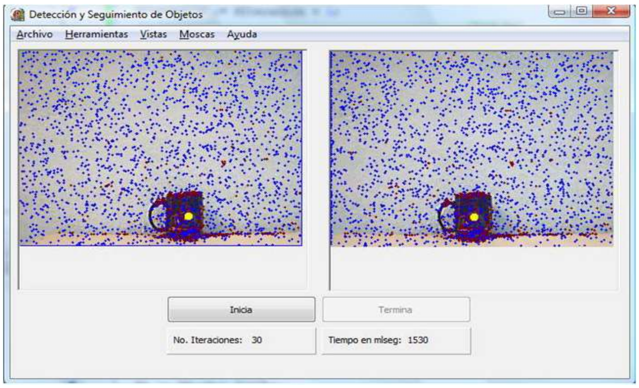

Fig. 8. Results after 30 executions using the first pair of images.

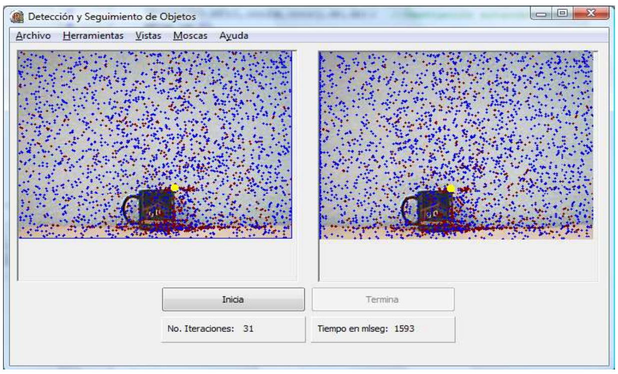

Fig. 9. The refresh of images, the flies are no longer posed on the object.

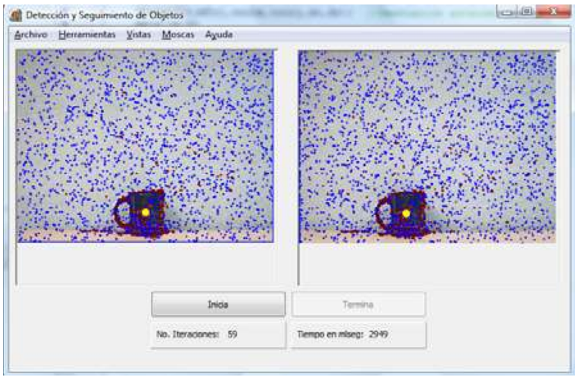

Fig. 10. Results after 59 executions using the second pair of images.

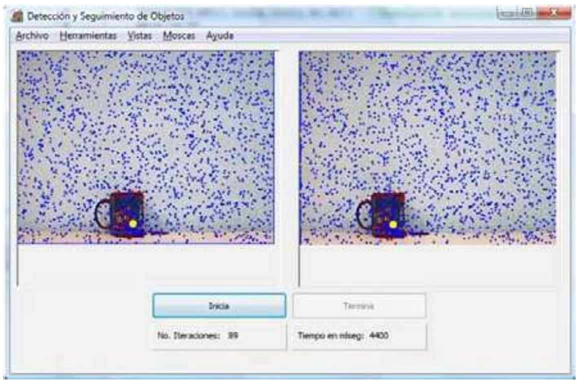

Fig. 11. Results after 89 executions using the third pair of images.

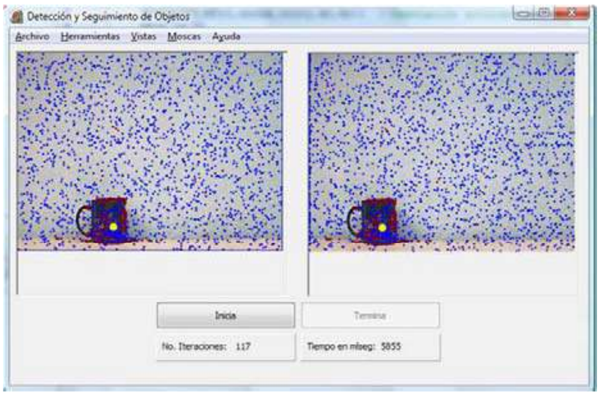

Fig. 12. The final result of the last pair of images after 117 executions.

8 Conclusion

We have presented a system for objects detection and tracking based on the fly algorithm and stereoscopy. We employ monocular vision that means only one camera is employed to capture the pair of stereo images. The camera displacement is carefully controlled. This system allows easy manipulation of all parameters involved in the process: the camera, the genetic algorithm parameters and the obtained centroid for the object detection and tracking. The system works very well and the object detection and tracking is obtained in real-time.

Finally, this system is being considered to be used in a mobile robot with monocular vision for navigation tasks or objects detection and tracking.

References

1. Álvarez, W.: Seguimiento de objetos para la reconstrucción de Ambientes en 3D. Universidad de Colombia (2007)
2. De Berg, M., Cheong, O., Van Kreveld, M., Overmars, M.: Computational Geometry, Algorithms and Applications. Heidelberg, Berlin, Springer, 3ª. Ed. (2008)
3. Blundell, B.G.: An Introduction to Computer Graphics and Creative 3-D Environments. Auckland, New Zealand, Springer (2008)
4. Boumaza, A.M. and Louchet, J.: Dynamic Files: Using Real-Time Parisian Evolution Robotics. In: EvoWorkshops (2001)
5. Chacón M.: Procesamiento Digital de Imágenes. México, Trillas (2007)
6. Gonzáles, R., Woods, R.: Tratamiento Digital de Imágenes. EUA, Addison Wesley (1992)
7. Hueso, D. l.: Visión por Computador. Fundamentos y Métodos. Prentice-Hall (2001)
8. Koza, J.: Genetic Programming II: Automatic Discovery of Reusable Programs. The MIT Press (1992)
9. Louchet, J., Guyon, M., Lesot, M., Boumaza, A.: Dynamic files: a new pattern recognition tool applied to stereo sequence processing. (2002)
10. Pajares, G., De la Cruz, J. M.: Visión por Computador: Imágenes Digitales y Aplicaciones. México, Alafaomega (2002)
11. Pauplin, O., Louchet, J., Lutton, E., Parent, M.: Obstacle Detection by Evolutionary Algorithm: the Fly Algorithm. Palmerston North, New Zealand (2004)