International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

Partial 3D Reconstruction using Evolutionary Algorithms

Mónica Pérez-Meza, and Rodrigo Montúfar-Chaveznava

Abstract**—**When reconstructing a scenario, it is necessary to know the structure of the elements present on the scene to have an interpretation. In this work we link 3D scenes reconstruction to evolutionary algorithms through the vision stereo theory. We consider vision stereo as a method that provides the reconstruction of a scene using only a couple of images of the scene and performing some computation. Through several images of a scene, captured from different positions, vision stereo can give us an idea about the three-dimensional characteristics of the world. Vision stereo usually requires of two cameras, making an analogy to the mammalian vision system. In this work we employ only a camera, which is translated along a path, capturing images every certain distance. As we can not perform all computations required for an exhaustive reconstruction, we employ an evolutionary algorithm to partially reconstruct the scene in real time. The algorithm employed is the fly algorithm, which employ "flies" to reconstruct the principal characteristics of the world following certain evolutionary rules.

Keywords**—**3D Reconstruction, Computer Vision, Evolutionary Algorithms, Vision Stereo.

I. Introduction

THE objective of this work is to present a method to carry out partial 3D reconstruction of a dynamic world in real time using the vision stereo theory, an evolutionary algorithm, and monocular vision. A set of algorithms that can give the solution to the vision stereo problem are the evolutionary and genetic algorithms. We specifically employ the fly algorithm [1], which was modified according to our conditions, tasks and objectives.

Using vision stereo is possible to obtain the depth of the objects in a scene. The capture of slightly displaced images of the scene can be accomplished for one of the following procedures:

- Aligning two or more cameras separated certain distance one of each other (similar to mammalian vision system).
- Displacing only one camera to several positions along a path and capturing images in the different positions.

In this work we use only one camera, applying geometry projective and modifying the fly algorithm to achieve partially the 3D reconstruction of the world. The camera is displaced to different controlled positions, capturing the images at each position and obtaining a 3D scene.

The results obtained in this work allows to considered the use of the system in mobile robot navigation, an activity where the robot needs to know in every moment the structure of the world to move around in a secure way.

II. Vision Stereo

A. The Pin-Hole Model

The pin-hole model camera is shown in Fig. 1. In this model, the three-dimensional point *P(x, y, z)* is projected on the image plan passing through the optic center in the focal plan. The straight line that links the point *P* and the optic center is the projection line and it intersects the flat image just in the pixel *p(x, y)* which is in the corresponding projection of *P(x, y, z)*. The optic center is located on the focal length of the flat image. This model is completed with the optic axis, which is a perpendicular line that begins in the center of the flat of image, passing through the optic center in the focal plan, and being perpendicular to *P(X, Y, Z)* as shown.

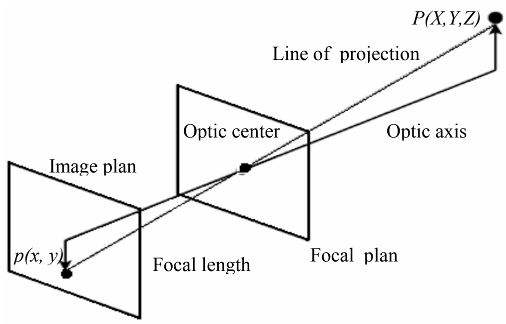

Fig. 1 The Pin-hole Model

Considering this model, and using geometry we have the equations that relates the spatial point *P(X, Y, Y)* to its corresponding projection point *p(x, y)* in the image:

$$x = f\frac{X}{Z} \text{ and } y = f\frac{Y}{Z} \tag{1}$$

B. Vision Stereo

In vision stereo or stereoscopic vision, we need two or more images to obtain the three-dimensional view. For example, if

Manuscript received September 29th, 2006.

Mónica Pérez-Meza is with the Computer Science School at the Benemérita Universidad Autónoma de Puebla, México (e-mail: arielmonp@yahoo.com).

Rodrigo Montúfar-Chaveznava is with the group GIRATE at ITESM Santa Fe, Av. Carlos Lazo 100, Col. Santa Fe, Del. Álvaro Obregón, 01389, México D.F. México (phone: +52 559177800; e-mail: rmontufar@itesm.mx).

we have two views of a scene slightly displaced one of another; both views have many characteristics in common; however each one contains certain visual information that the other does not have. The difference presented between both images is called disparity.

The way of perceive the sensation of depth, distance or nearby of the objects that surround us, is accomplished by a process of fusion which is called stereopsis. The mechanisms of the stereopsis are the following:

- If we observe very distant objects, the optic axes of our eyes remain parallels.
- If we observe a nearby object, our eyes rotate so that the optic axes converge on certain point. At the same time, an accommodation is produced or focuses to could see clearly the object. This joint process is called fusion.

III. Evolutionary Algorithm Characteristics

Evolutionary algorithms manipulate individuals, which are evaluated by a function, called fitness function, in analogy to the biological evolution. The general diagram of the evolutionary algorithm is presented in Fig. 2.

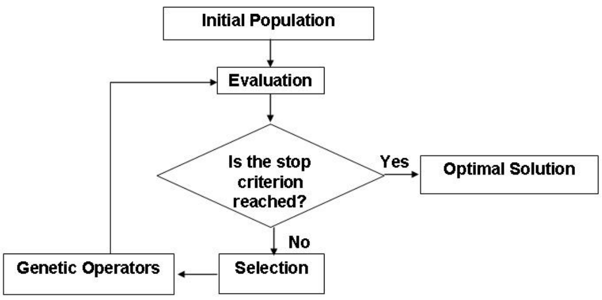

Fig. 2 General Layout of Genetic Algorithms

The principal characteristics are:

- The population is a group of individuals.
- An individual is defined by its genes $X = (x_1, x_2,\ldots,x_n)^T$, which usually represents its coordinates in the search space.
- The evaluation is the computation of the fitness value in every individual.
- The selection eliminates part of the population, keeping the best individuals according the previous evaluation.
- The evolution applies genetic operators (crossover, mutation, etc.), leading to new individuals in the population.

Some kinds of evolutionary algorithms are:

- Genetic algorithms, which are a technique of programming that imitate to 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 instructions in a programming language.

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.

A. *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 a flat image. The evolution is regulated by a function of adjustment designed in such a way that the flies converge on a physical object located in the scene. This function is called the fitness function.

The fly algorithm uses individual evolutionary strategies for the analysis of stereo images. The algorithm considers points in the 3D space as primitives and evolves all individuals using the fitness function and evolutive rules. The purpose is to concentrate all the flies on the surface of some objects of the scene.

A fly is defined as a 3D point with world coordinates *(x,y,z)*. The flies are projected over two images, producing a couple of new coordinates: (x_R, y_R) for the right image and (x_L, y_L) for the left image.

At the beginning, the population of flies is created randomly in the intersection area of the view of both images.

The fitness function

The fitness function used to evaluate a fly compares its projections on the left a right images provided by the cameras. If a fly is settled on an object, the projections will have similar pixel neighborhoods on both images and then this fly will have a high fitness value. This idea is illustrated in Fig. 2 and 3. Fig. 3 shows the neighborhoods of two flies on left and right images. In this example, *fly1*, which is settled on an object, has a better fitness value than *fly2*.

The fitness function is [2, 3]:

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

$$G = \left|\nabla(M_L)\right|\left|\nabla(M_R)\right| \quad (3)$$

where:

- (x_L, y_L) and (x_R, y_R) are the coordinates of projected individual on the left and right images respectively.
- $L(x_L + i, y_L + j)$ and $R(x_R + i, y_R + j)$ are the pixel color values in the left and right images respectively.
- 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 left and right projections of the fly. That is intended to penalize flies when they are settled on uniform regions, i.e. the less significant flies.
- In the color images, the difference of squares is calculated on each channel of color. We have only one channel for gray images.

Selection

Selection is elitist and deterministic. It classifies flies according to their fitness values and keeps the best individuals (around 40% or 50%).

A sharing operator [2, 3] reduces the fitness of flies packed together and forces them to explore other areas on the search space.

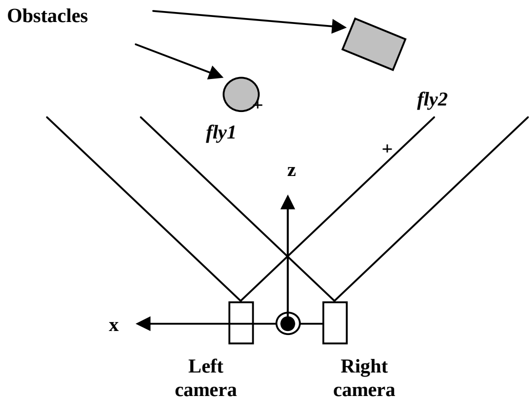

Fig. 3 View of two flies in the neighborhood

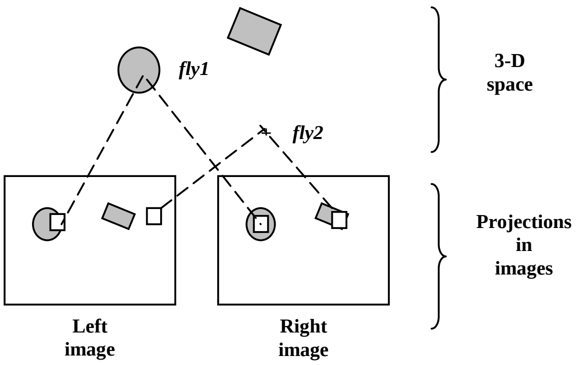

Fig. 4 Flies projections on the left and right images

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:
$$\overrightarrow{OF} = \lambda\overrightarrow{OF_1} + (1-\lambda)\overrightarrow{OF_2}$$
where λ is a random value between [0, 1].
- **Gaussian mutation**: This operation adds Gaussian noise to each component of the 3D coordinate of the mutated fly.
- **Immigration:** This operation is used to extend the exploration area in search space, creating new individuals randomly. It ensures a constant exploration of whole the search space.

Edge detection

Edge detection techniques are used to locate points where a sharp intensity variation is presented. The operators of these techniques can be applied partially or over the entire image. The basic solution employed by many edge detection algorithms is the computation of local differential operators. In this work we employed the Sobel operator, which is based on the first differential.

Morphological operators

In artificial vision, it is frequent to use the morphology for the processing of regions to determine their shape or changes. In this work, we employ some common morphological operations: dilation, erosion, opening and closing.

Dilation (⊕) is a morphological transformation that combines two sets using the addition of vectors (Minkowski's addition). The dilation of X by B, denoted $X \oplus B$, is defined as:

$$X \oplus B = \left\{ d \in E^2 : d = x + b,\ \forall x \in X \ \ \forall b \in B \right\} \quad (4)$$

Erosion (⊗) is a morphological transformation that combines two sets using the subtraction of vectors. It is the dual of the dilation. Erosion and dilation are not inverse transformations.

$$X \otimes B = \left\{ d \in E^2 : d = x - b, \forall x \in X \ \ \forall b \in B \right\} \quad (5)$$

Opening is a two steps operation: the erosion followed by the dilation. In general, this operation smoothes the contour of an image, breaking out narrow isthmuses and eliminating thin protuberances.

Closing is performed carrying out the dilation operation followed by the erosion operation. This operation tends to smooth sections of contours but, in opposition to the opening, generally fuses narrow separate and thin coming and deep, it eliminates small holes and fills holes of a contour.

IV. The Three-Dimensional Reconstruction System

The proposed system comprises eight modules: Open, Save, Exit, Calibration, Regions Size, Options, Views and Reconstruct. The Reconstruct module contains five sub-modules: Fitness function, Correspondence, Crossing, Mutation and Renew.

We present the system architecture in Fig. 5. All modules are accessed from the main menu.

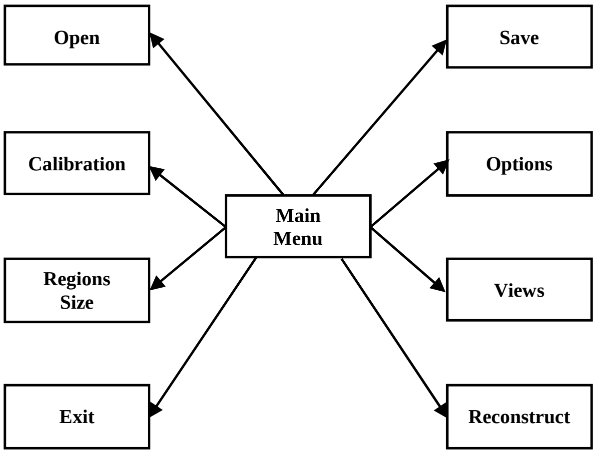

Fig. 5 The system Architecture

Next, we describe the system modules:

a) Open. It is for opening a new file of image.

b) Save. Allows saving an image that is shown on screen.

c) Exit. This module is to exit the system.

d) Calibration. External and internal camera parameters are introduced in this module.

e) Regions Size. In this module the regions sizes to divide the images are introduced.

f) Options. The population size is defined in this module.

g) Reconstruction. This is the most important module. It performs the 3D reconstruction form the 2D images. It has the following modules.

i) Fitness function. The fitness function (2) is applied to the population of flies in this module. If the color differences between each fly and its neighborhood are low, almost zero, the value of the fitness function is high for this fly.

ii) Correspondence. In this module we get the correspondence of the flies in the right and left images.

iii) Crossing. This module allows generating a new fly from two parents using the crossing function.

iv) Mutation. In this module a new fly is obtained by adding Gaussian noise to a parent fly.

v) Renew. New flies are generated with this module.

The Reconstruct architecture is shown in Fig. 6. We note that to carry out the reconstruction, the fitness function is applied every time, so we can select the best flies and delete the worse ones. The mutation, crossing and renew operators are applied at different percentages of population previous to be projected in the right and left images.

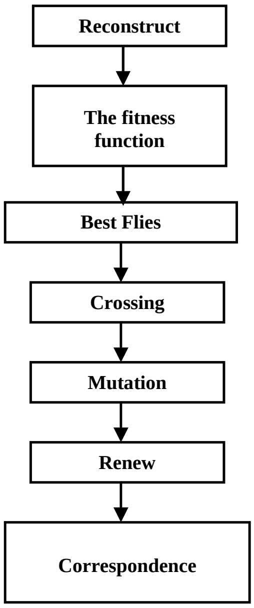

Fig. 6 3D Reconstruct Module Architecture

V. RESULTS

The results were obtained with a population of 3000 flies, using five images displaced five centimeters on horizontal axis one each one other. The left and right images are refreshed every two seconds approximately. The process is presented in Fig. 7. The parameters employed were:

Flies to preserve every generation: 50%
New flies by crossing: 20 %
New flies by mutation: 20%
New flies by random operation: 10%
Region size for control: 160 x 120 pixels

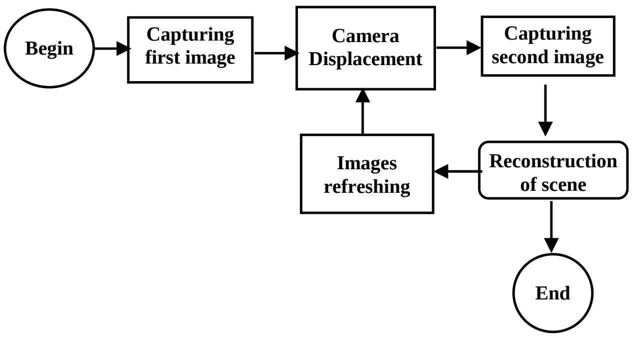

Fig. 7 The 3D reconstruction process

Fig. 8 shows to the left and right images. The initial population of flies is also presented. The images are divided in four regions, each with 750 flies. Next figures. present the execution process.

After 15 executions, the flies are located on the objects as presented in Fig. 9. Next the camera is translated five centimeter on the horizontal axis, then the left image is discarded, the right image takes the place of the left image and a new image is placed in the location corresponding to the right image. This new disposition is shown in Fig. 10. We can observe how the flies are not placed on the objects.

Fig. 11 shows the result of the second couple of images after 29 executions. The flies are posed on the objects again.

From Fig. 12 to 15 the process described above is repeated and we note how the flies are looking for the objects to pose on them. The flies' algorithm is working very well.

Every fly is a point in the world with coordinates *(X, Y, Z)*. We fusion all flies to obtain the 3D representation of the world. However, there are some holes in the reconstruction and some isolated flies, then, only for visualization, we apply the morphological operators to get better structures of the environment. Fig. 16 shows the view of the *X-Y* plan after applying the opening and closing operators.

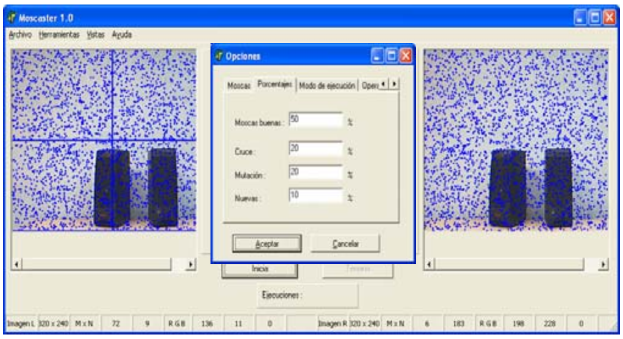

Fig. 8 Division of the left image in four regions with 750 flies by region

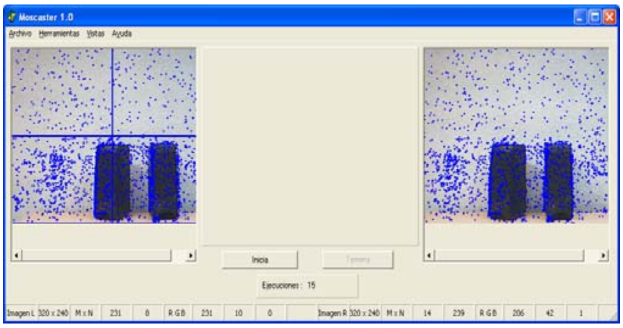

Fig. 9 Results after 15 executions using the first pair of images

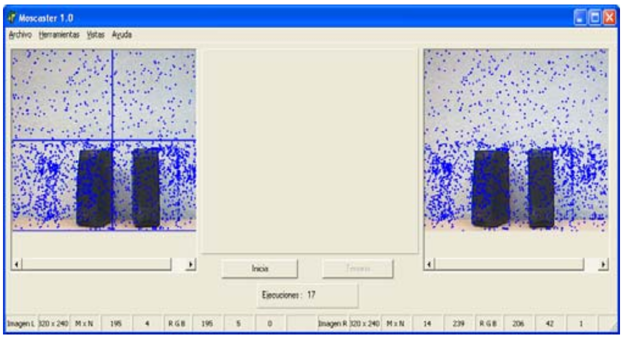

Fig. 10 The refresh of the images, suddenly the flies are not posed on the objects

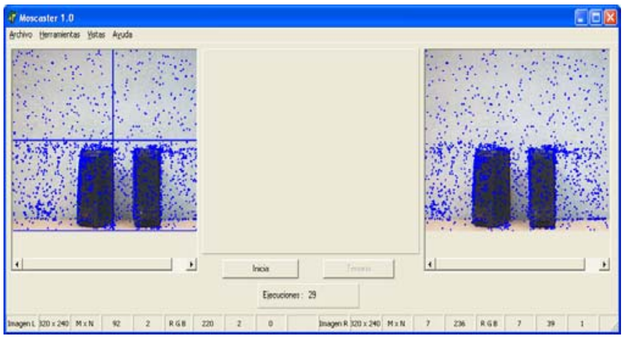

Fig. 11 The result of the second couple of images after 29 executions

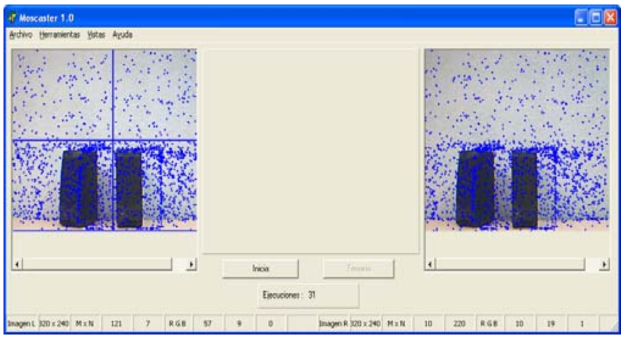

Fig. 12 A new refresh of images

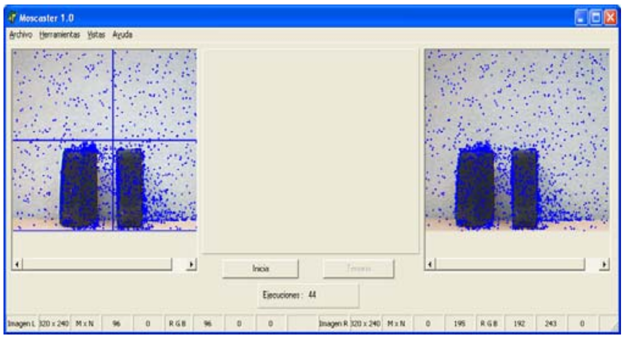

Fig. 13 The result of the second couple of images after 44 executions

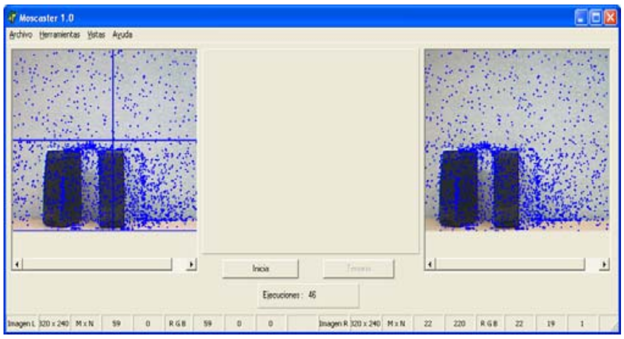

Fig. 14 Another image refreshing

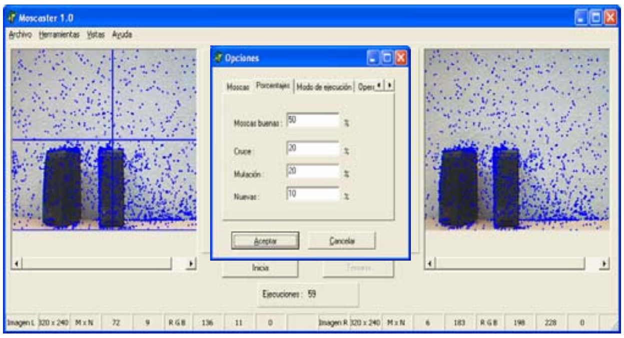

Fig. 15 The final result of the last couple of images after 59 executions

Fig. 16 The *X-Y* view. We apply the opening operation to deleted isolated flies, next we apply the closing operation to delineate the objects

VI. Conclusion

We have presented a system for partial 3D reconstruction based on the fly algorithm. We employ monocular vision instead of stereo vision; this means only one camera is used. The camera displacement is carefully controlled.

The system allows manipulating easily all parameters involved in the process: the camera and the genetic algorithm parameters. The system works very well and the 3D reconstruction is obtained in real-time.

Images are divided in regions to disperse the flies on the entire image plan. In this way, we avoid an excessive concentration of flies in certain points.

Finally, we perform an opening and closing morphological operations to better appreciate the reconstruction.

This system is considered to be used in a mobile robot with monocular vision for navigation tasks.

References

[1] Louchet, J. September 2000. “Stereo analysis using individual evolution strategy”. International conference on pattern recognition. Barcelona.

[2] Boumaza, A. M., Louchet, J. 2001. “Dynamic Flies: Using Real-Time Parisian Evolution in Robotics”. EvoWorkshops, pp.. 288-297.

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