Active contour: a parallel genetic algorithm approach

Florence Kussener[1]

[1] MathWorks,
2 rue de Paris
92196 Meudon Cedex, France
Florence.Kussener@mathworks.fr

Abstract

This paper presents an algorithm for automatically detecting contours using snake algorithm. Prior knowledge is first used to locate initial contours for the snakes. Next we optimize the energy by using genetic algorithm approach. We introduce parallel computing to reduce computation time for the genetic algorithm calculations.

Key words

Active contour, parallel computing, genetic algorithm, segmentation, snake

1 Introduction

Edge detection is a fundamental tool in image processing, image pattern recognition, and computer vision techniques. Ideally, the result of applying an edge detector to an image should lead to a set of curves that indicate the contour of objects as well as curves that correspond to discontinuities in surface orientation. Furthermore, applying an edge detection algorithm to an image may reduce the amount of data to be processed while preserving the structural properties of an image. Edge detection is not a trivial task. There are many methods for edge detection. In this paper, we will introduce the active contour method, or snake algorithm, which minimizes the energy function.

Active contours have multiple advantages over classical feature attraction techniques. Snakes are easy to manipulate using external image forces. They are self-adapting in their search for a minimal energy state. Furthermore they can be used to track dynamic objects in temporal as well as the spatial dimensions.

Nevertheless, one of the biggest drawbacks of this method is that snakes are get stuck in local minima states. This may be overcome by using genetic algorithm techniques at the expense of longer computation times.

Genetic Algorithm is an optimization solver, which does an analogy to Darwin evolution by combining mutation, crossover and selection step. One of the biggest advantages of Genetic Algorithm is its ability to find a global optimum. A consequence of this search is to have an additional computation time. This is why Parallel Computing is a good approach to optimize time of computation.

Parallel Computing is the use of two or more processors in combination to solve a single problem.

2 Active Contour

Several methods are existing for edge detection in images. The most known algorithms are Sobel and Prewitt methods. These methods are calculating the gradient at each points giving the direction of the largest possible increase from light to dark and the rate of change in that direction. A default of these methods is to detect every small objects but not the complete contour of the general shape (Fig 1).

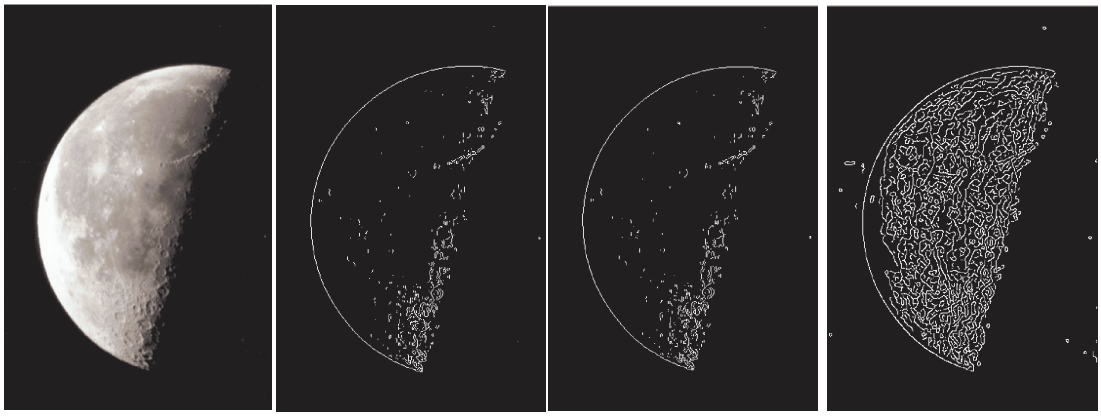

Figure 1: (a): initial image, (b): sobel filtering, (c): prewitt filtering, (d): canny filtering

The Snake or active contour model, is a method for delimiting an object edge from a possibly noisy 2D image. This method attempts to minimize energy associated to the current contour as a sum of internal and external energy. The internal energy is supposed to be minimal when the contour is at the object boundary position. The most usual approach consists in giving low values when the regularized gradient around the contour position reaches its peak value. The external energy should be minimal when the snake has a shape which is relevant considering the shape of the object. The simplest approach assigns high energy to elongated contours and to high curvature contours, the shape should be as regular and smooth as possible.

Snake is an "active" model as it always minimizes its energy function and therefore exhibits dynamic behavior.

A simple elastic snake is thus defined as

- a set of n points
- an internal elastic energy term
- an external edge based energy term

A snake has to be initialized near the target and it will iteratively be attracted towards the salient contour. If we define a snake by a collection of *n* points, *vi=(xi,yi)*; where *i=1..n,* then the energy function will be defined by the following equation

$$E_{total} = E_{internal} + E_{external} \quad (1)$$

where $E_{internal}$ represents the internal energy of the contour and $E_{external}$ represents the external energy acting on the snake. In general, the Snake is placed near the object contour. It will dynamically move towards object contour by minimizing its energy iteratively.

Each iteration can be presented by the following steps:

- Internal and external energy computation, according to the position of the points.
- For each point, computation of a new position, minimizing the energy

While Stop Criteria
  For all active contour points
    For all proximity points
      Energy computation
    End - For
    Minimization to select the best point
  End - For
End - While

All the programming has been implemented in MATLAB (Fig 2).

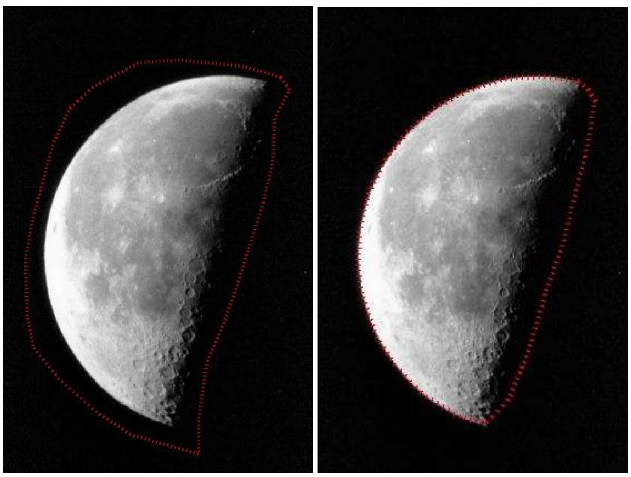

Figure 2 : on left side, initial path, on right side, optimal path, after 35 iterations

We notice that it can be useful to add or suppress points at each iteration (for example when points are too far apart).

Internal energy is not dependant on the image or of the shape of the object. It is only dependant on contour point (curve, etc). In fact, the contour has to keep a round shape by minimizing the derivative (first, second orders) and has to avoid a point that is too distant. Ideally, an internal energy is minimized for a circle with regularly spaced points.

$$Einternal = c\_int * (\alpha * \sum_{i=1}^{n} \nabla I(xi, yi) + \beta * \sum_{i=1}^{n} \nabla(\nabla I(xi, yi))) \quad (2)$$

During the test, we define $\alpha = \beta = 1$.

External Energy is dependant on the contour impact on the image. We have to consider the opposite of the gradient value in each point. This energy is minimal when the snake is exactly on the shape.

$$Eexternal = c\_ext \sum_{i=0}^{n} U(xi, yi) \quad (3)$$

U is the image energy, defined by the complement of the rotationally symmetric Laplacian of Gaussian filter of size 5 with a standard deviation of 0.5 (Fig 3).

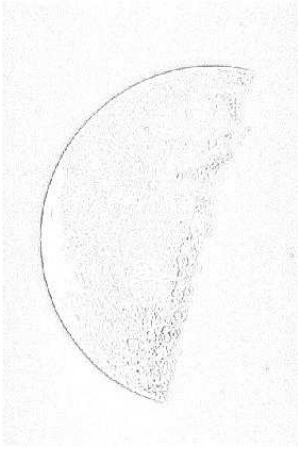

Figure 3: Energy of the image

The definition of parameters for different energies is a delicate step and is dependant on the shape of the curve (Fig 4). Internal energy will be penalized in case of concavity. In this case it may be useful to add a parameter search step.

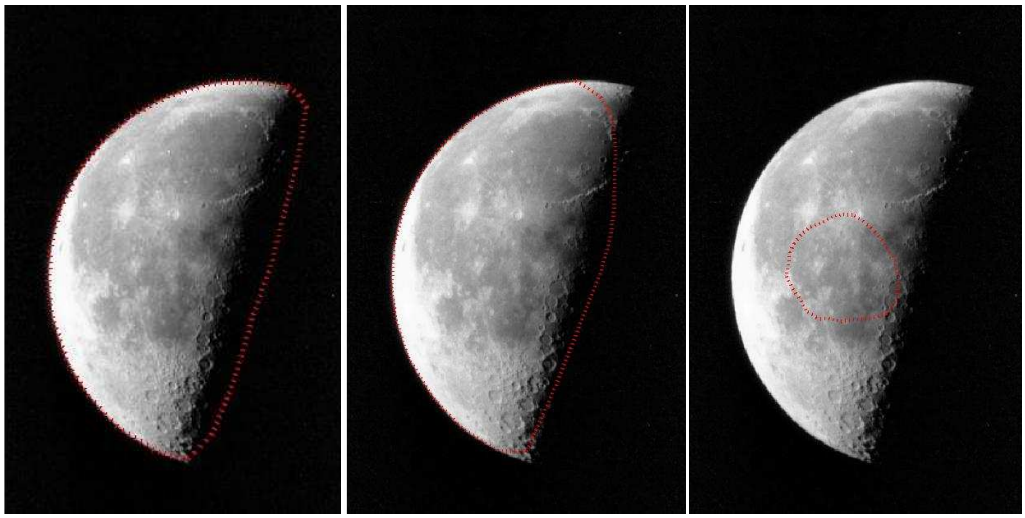

Figure 4: Results obtained with the same initial path, and c_ext=1. On the left side, c_int=0.05, on the central image; c_int=0.2, on the right image: c_int=0.5,

In our case, we consider c_int=0.05 and we do not process a parameter scanning to find the optimal parameter depending on the concavity of the shape to have a generalized approach. Note that this search could be in parallel on different core.

We also observed that, depending on the initial path (number of points, localization of points), the result is not always optimal (Fig 5). This is due to the fact that the active contour method is dependant on the initial snake. Different approach could have been choosing: add initial step to choose the initial path [2] or use meta-heuristic algorithm such as simulated annealing [3] or genetic algorithm [4]. In our case, we decided to use a genetic algorithm approach which should be really beneficial. One advantage of genetic algorithm compared to simulated annealing is the fact that parameters are easier to define.

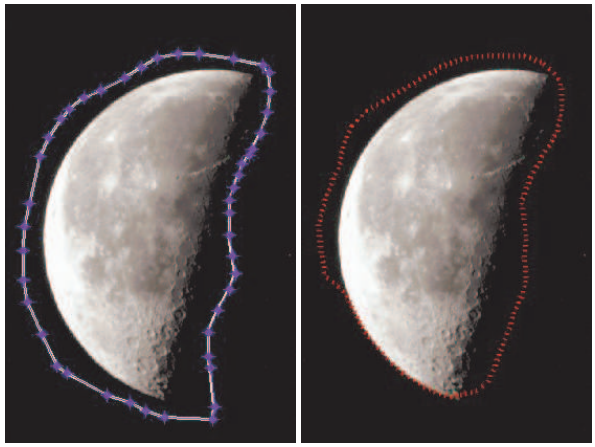

Figure 5: on left side, initial path, on right side, optimal path, after 34 iterations

3 Genetic Algorithm Optimization

We discussed snake algorithm. The objective of this algorithm is to minimize the energy function defined by Eq(1). The major problem of this method is to be attracted by a local minimal. It is why the use of a genetic algorithm is a good approach to find the global optimum snake.

3.1 Genetic Algorithm Introduction

Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. It belongs to the larger class of evolutionary algorithm, which generates solution to optimization problems using natural evolution, such as mutation, selection and crossover. Initially developed in biology context, such methods find application in bioinformatics, phylogenetics, economics, chemistry, manufacturing, physics, and other fields.

A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate the solution domain. Once these functions are defined, GA proceeds to initialize a population of solution randomly, them improve it through repetitive application of mutation, crossover and selection operators.

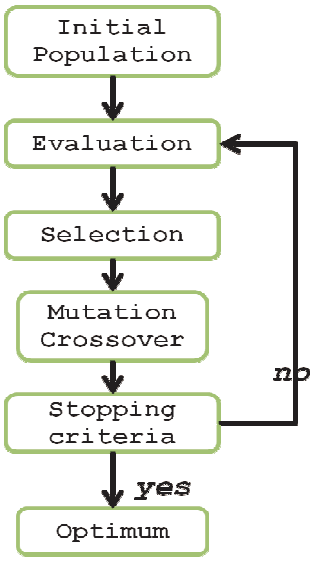

Figure 6: Genetic Algorithm steps

Figure 6 illustrates the different steps to obtain a minimum:

- Initialization: initially many individual solutions are randomly generated to form an initial population. The population contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions.
- Selection: During each successive generation, a proportion of existing population is selected and conserved for the new generation
- Crossover: A pair of parents is selected for producing a child solution. The new generation shares many of the characteristics of its parents.
- Mutation: a parent solution is selected for producing a child solution. The new generation shares the same information than its parents, with random modifications.

Although Crossover and Mutation are the main genetic operators it is possible to use other operators, such as regrouping, colonization- extinction or migration in genetic algorithm.

The previous steps are repeated until a termination condition has been reached: a solution is found that satisfies minimum criteria, fixed number of generation reached, allocated budget reached, etc.

3.2 Genetic Algorithm Implementation

We have to optimize the energy function. For this paper, we use MATLAB software and the Global Optimization toolbox which provides a genetic algorithm solver. This genetic algorithm solver can be customized and we can define objective function, initialization function, crossover and mutation function. The selection step is directly defined by MATLAB. We just have to specify a set of parameters. We also provide parameters for termination step. In the following, we will provide further details.

3.2.1 Initialization

From an initial snake (not a good candidate for optimal solution in active contour case), we add some noise to create other snakes (Fig 7). In our tests, we use 300 snakes to implement the genetic algorithm. Note that genetic algorithm convergence is dependant on the number of snakes.

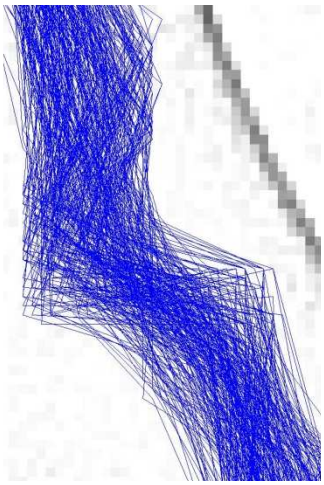

Figure 7: initial snakes for 300 individuals

3.2.2 Selection

The selection step chooses parents for the next generation based on their scaled values from the fitness scaling function. We use the stochastic uniform selection method which lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size. We guarantee that 2 individuals will survive to the next generation.

3.2.3 Crossover

Crossover operator generates two children from two parents selected by the genetic algorithm solver. We define a crossover at one point. We choose a random integer n between 1 and the length of the snake and then

- Selects vector entries numbered less than or equal to n from the first parent.
- Selects vector entries numbered greater than n from the second parent.
- Concatenates these entries to form a child vector.

For example, if `p1` and `p2` are the parents

```
p1 = [a b c d e f g h]
p2 = [1 2 3 4 5 6 7 8]
```

and the crossover point is 3, the function returns the following child.

```
child = [a b c 4 5 6 7 8]
```

The fraction of the next generation that crossover produces, is set at 0.8. Mutation produces the remaining individuals in the next generation.

3.2.4 Mutation

The mutation function applies the same workflow as the active contour method, but only on the selected parents.

3.2.5 Termination

We specify a limit of 1000 generations. Furthermore, the algorithm stops if there is no improvement in the objective function for 50 consecutive generations.

3.3 Summary

Genetic Algorithm gives us good results compared to the initial approach. In addition to converge to an optimum, it helps us with parameter selection. For example, in Figure 8, we can observe that the active contour method has difficulty to find the contour. This is due to the fact that the initial contour has a higher number of points than previous trials and Internal Energy is too high a value compared to the External Energy: when we change the position of one points, the consequence is high on internal energy value.

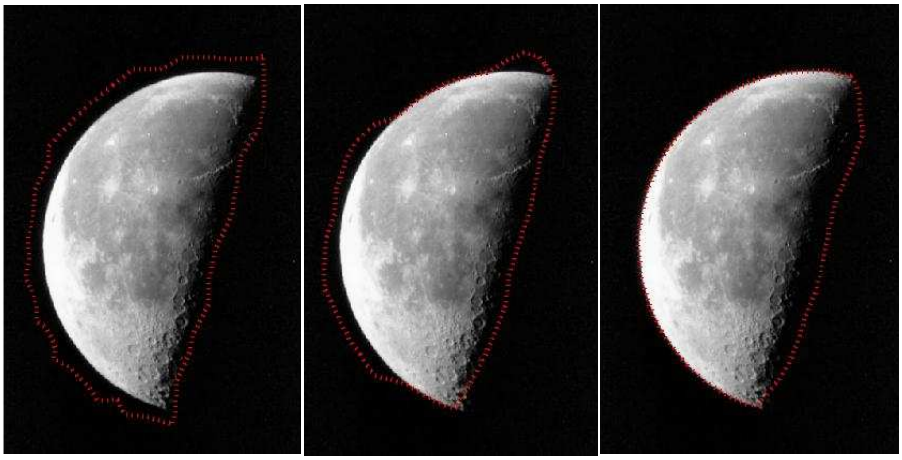

Figure 8: on left side, initial set of points. Central image: active contour algorithm. Right image : Genetic Algorithm Active Contour.

4 Parallel Computing

Parallel Computing is a form of computation in which many simulations are carried out simultaneously, based on the principle that large problems can often be divided into smaller ones, which are then solved in parallel. There are different approaches to parallelize algorithms. In our case, the same calculation will be performed on different sets of data. We will see how to increase computation performance on GA.

4.1 Parallel Computing Environment

Since 2007, MATLAB has a toolbox, named Parallel Computing Toolbox. This toolbox uses MPI functionality to parallelize application. The main advantage to use this toolbox is to develop parallel application with no so much time of development and with very few modification of the code.

Specifically, the Parallel Computing Toolbox proposes a PARFOR loop which can run independent iterations on the different core activated by the configuration. The configuration can be local or on a cluster for large simulations. The biggest limitation of this parfor loop is to have independent iterations. This is simple to check: if each iteration can be done in another order, we can conclude that the FOR loop can be changed as a PARFOR loop.

Technically, MATLAB creates a function with the body of the PARFOR loop, launch it on the selected core and get back the output of the function at the end of the simulation. Even if MATLAB uses advanced technology, as MPI, for this process, it is really simple to model it with this high level functionality.

4.2 Parallel Genetic Computing

In our case, we use a PARFOR loop at crossover and mutation function level. Instead of computing the next generation, child by child, we do it in the same time on the different cores of the computer.

4.3 Experimental Results

We run the parallel genetic algorithm on different configurations from 1 core to 8 cores. We set the population size to 300. For each configuration, we completed4 runs. Table 1 collapses the different results.

	Time of Computation (in seconds)			
Genetic Algorithm Active Contour	110.49	93.91	100.93	108.74
Parallel Genetic Algorithm Active Contour (2 cores)	74.89	97.51	86.26	53.48
Parallel Genetic Algorithm Active Contour (4 cores)	58.51	53.53	43.70	35.99
Parallel Genetic Algorithm Active Contour (8 cores)	46.51	39.43	36.30	39.31

Table 1: Results of the simulation on different configuration

Another way to see the results is a graphic (Figure 9) so we can easily see the trend and imagine which would be

the result with bigger configuration.

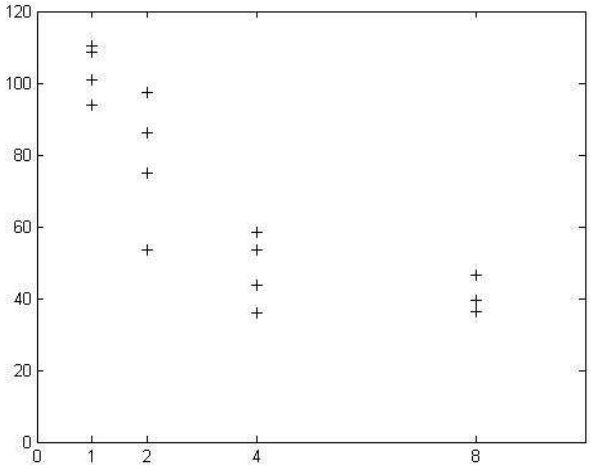

Figure 9: Time of execution depending number of cores

Due to the fact that the genetic algorithm is a heuristic algorithm and has random evolution, we observe that time of convergence is not repeatable and we can observe a large spread of time of execution with a significant gain as the number of cores grows.

5 Conclusion

The active contour has multiple advantages over classical feature attraction techniques. Nevertheless, one of the largest drawbacks of this method is that snakes often get stuck in local minima states. This has been overcome by using genetic algorithm techniques at the expense of longer computation time. To minimize this execution time, we used parallel computing. The use of multiple processors in combination gives us good results and we could project that this would be even more efficient if we use a cluster of CPUs.

References

[1] LAKHADARI M.., *Segmentation d'images par contour actif en appliquant les algorithmes génétiques*. PhD thesis, INI

[2] LEAN CCH., SEE AKB., ANANDAN SHANMUGAM S., *An enhanced Method for the Snake algorithm.* Innovative Computing information and control 2006

[3] LIQUN T. KEJUN W. GUANGSHENG F. YONGHUA L., *An image segmentation algorithm based on the simulated annealing and improved snake model.* Mechatronic and automation 2007.

[4] TAN JH., NG* EYK., ACHARYA R., *Detection of eye and cornea on IR thermogram using genetic snake algorithm.* 9th International Conference on Quantitative InfraRed Thermography.