Boundary Extraction Through Gradient-Based Evolutionary Algorithm

Román Katz and Claudio Delrieux
Departamento de Ingenieria Electrica y de Computadoras
Universidad Nacional del Sur - Av. Alem 1253, (8000) Bahia Blanca, ARGENTINA

ABSTRACT
Boundary extraction is an important procedure associated with recognition and interpretation tasks in digital image processing and computer vision. Most of the segmentation techniques are based on the detection of the local gradient, and then their application in noisy images is unstable and unreliable. Therefore global mechanisms are required, so that they can avoid falling into spurious solutions due to the noise. In this paper we present a gradient-based evolutionary algorithm as a heuristic mechanism to achieve boundary extraction in noisy digital images. Evolutionary algorithms explore the combinatory space of possible solutions by means of a process of selection of the best solutions (generated by *mutation* and *crossover*), followed by the evaluation of the new solutions (*fitness*) and the *selection* of a new set of solutions. Each possible solution is in our case a contour, whose *fitness* measures the variation of intensity accumulated along it. This process is repeated from a first approximation of the solution (the *initial population*) either a certain number of generations or until some appropriate halting criterion is reached. The uniform exploration of the space of solutions and the local minima avoidance induce to form better solutions through the gradual evolution of the populations.
Keywords: Boundary Extraction, Pattern Recognition, Image Processing, Evolutionary Algorithms, Metaheuristics.

1 Introduction

Boundary extraction is an important procedure for segmentation and pattern identification purposes in digital images, not only for recognition and interpretation tasks but also for object classification [4]. The gradient operator is a widespread tool used for these purposes, detecting local level variations that could correspond to contours of interest. Although this methodology provides acceptable results for typical cases, there are a number of situations in which an extra computational effort is required to expand the possible range of successful application. This is the case of boundary extraction in noisy images or with non-uniform intensity levels objects. Under these conditions the common applied strategy consists on a border detection procedure by means of the gradient operator followed by some kind of global processing. One of the modern techniques most applied in this sense -and under permanent research- is the *active contour* approach (also called snakes) [1, 6]. It consists on the utilization of user-initialized curves that "evolve" inside the image until they find the contour sought for, taking advantages of different possible mechanisms, as B-splines, vector gradient flow and others. In general the active contours have limitations regarding the concavities of the contour to segment [10].

In this work we propose a boundary extraction system that combines the use of the gradient operator within the implementation of an evolutionary algorithm. Evolutionary algorithms [2] are a means for finding nearly optimal solutions to non-trivial and computationally expensive problems that have a very precise mathematical formulation. For this reason in this work we will recast the boundary extraction problem as an instance of an combinatory optimization problem. In Section 2 we present the main concepts of boundary extraction, with particular emphasis on noisy images and with non-uniform intensity levels. In Section 3 we introduce the idea of evolutionary algorithms and their application to boundary extraction in our particular case. We finally present in Sections 4 and 5 respectively, some illustrative examples and the conclusions.

2 Boundary Extraction

The most widely used feature extraction techniques characterize the contours of an object through the detection of its boundary, according to local ostensible discontinuities in the intensity levels $I(x, y)$ with respect to neighboring pixels [4]. It is adequate then, to use a gradient operator to find these local intensity changes. This gradient is mathematically described as [3]:

$$\nabla I(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix}. \quad (1)$$

In many occasions it is very convenient to regard in particular the *scalar gradient* $\nabla = \sqrt{G_x^2 + G_y^2}$ which is independent from the direction of the gradient itself. We will refer to ∇ as the scalar gradient if not stated otherwise. The operators G_x and G_y represent generic implementation of the directional gradients of the digital image, and could be easily obtained through the Roberts, Prewitt or Sobel "masks". We have adopted for this work the Sobel mask because it achieves the best results. If $I(x, y)$ denotes the intensity in the pixel (x, y), then the utilization of the mentioned operator in the pixel (x_i, y_j) is:

$$G_x = (I(x_{i-1}, y_{j-1}) + 2I(x_i, y_{j-1}) + I(x_{i+1}, y_{i-1})) -$$

$$
\begin{aligned}
& (I(x_{i-1}, y_{j+1}) + 2I(x_i, y_{j+1}) + I(x_{i+1}, y_{i+1})) \\
G_y \;=\; & (I(x_{i-1}, y_{j-1}) + 2I(x_{i-1}, y_j) + I(x_{i-1}, y_{j+1})) - \\
& (I(x_{i+1}, y_{j-1}) + 2I(x_{i+1}, y_j) + I(x_{i+1}, y_{j+1}))
\end{aligned}
$$

Therefore, the operators could be found through the respective convolution of the image with the scalar masks

$$
\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.
$$

We present in Fig. 1(b) the graphic of the scalar gradient ∇ of the image shown in Fig. 1(a). In this image the main object has a uniform intensity level and there is no noise. For this reason, a threshold of the scalar gradient renders a highly perceptible boundary for the object. However, in noisy images this segmentation technique will fail because of spurious blurts of the gradient above any useful threshold level. For instance in Fig. 1(c)), the magnitude of the scalar gradient ∇ still captures the contour for a human assisted segmentation (see Fig. 1(d)), though considerably degraded as a consequence of the "amplification" of the small local perturbations produced by the noise. Any kind of segmentation for automatic recognition purposes is doomed to fail.

Regrettably, the presence of noise in digital images is almost unavoidable, in most cases due to the acquisition process or to numerical and quantization errors in the storing/processing stages. Moreover, the normal situation in real applications (v. g. robotic vision) adds varying intensity levels and not uniform illumination conditions in the objects to be recognized. If we also consider that the objects within an image may have different and varying intensity levels (Fig. 1(e)), then the contour detection through the gradient ∇ is even worse (Fig. 1(f)).

Then, we can assume that segmentation by means of local processing techniques based on the isolated application of the gradient operator is very restricted. This operator amplifies the small local differences in the noisy images and the extraction process tends to deviate from the optimal solution. Feature extraction is mostly a global operation. Any successful strategy under these conditions combine contour detection and global processing schemes to avoid local minima. This is in fact the of the most known global extraction mechanisms, as the Hough transform for border detection, graph search methods [3] and dynamic programming [4]. Here we propose a different approach. Perhaps a good measure of the adequacy of a proposed boundary may be the accumulated intensity difference through successive pixels. In this setting, a gradient operator minimizes this difference locally, but may remain stuck in local minima. However, a non-local operator may overlook these local minima and search for better global solutions. Therefore, the extraction problem is recast as an optimization problem that can be solved with heuristic search, as we will see in the next Section.

3 Evolutionary Algorithms

3.1 Preliminary Concepts

Evolutionary algorithms are stochastic search methods that allow an exploration of a large space of solutions with the objective of finding a solution satisfactorily close to the optimal in an acceptable time. Among these kind of algorithms are evolutionary programming (EP) [9] and genetic programming (GP) [5], which have both a common origin in the imitation of natural evolution. EP and GP are adequate to solve problems that are either impractical or unmanageable through traditional Artificial Intelligence techniques (heuristic search, logic, etc.). EP and GP are successfully employed in several problems, including combinatory optimization, scheduling, classification, system identification and pattern recognition [7, 8].

The main idea derives from a metaphor of natural evolution in biological processes, where the individuals (phenotypes) express a genetic information (genotype) and are subjected to rules of "survival of the fittest" (*fitness* and *selection*). Those best fit individuals (with higher fitness) have more chances of survival and generate their offspring, which may be subject of further processes of *mutation* and *crossover*. This general scheme has been used with variations in many successful different applications.

An evolutionary algorithm maintains a *population set* or generation of possible solutions of the problem and allows them to progress through the transformation of the *population* in the successive generations. The transformation is produced by *mutation* and *crossover* operations, by means of which the population set of the next generation is assembled from the actual one. The *crossover* operator combines the genotype of two or more solutions to generate a new genotype, and the *mutation* operator generates a new genotype as a random perturbation of the genotype of a previous solution. In each of these generations a *fitness* function (related in some way to a cost function) is evaluated, in order to quantify the adequacy of every individual. Therefore, there exists a *selection* stage that chooses the best fit individuals to conform the new generation.

This process of mutation, crossover, fitness evaluation and selection is repeated from a first generation (*initial population*) either until a certain number of generations were produced or until some appropriate halting criterion is reached. The first generation may be any suitable approximation of the solution sought for. The uniform exploration of the space of solutions and the avoidance of local minima (mostly due to the effect of the mutation operator) induce to refine progressively better solutions. Finally, the best fit individual at the last generation or the best fit individual ever observed is chosen as the (so far) best solution.

3.2 Evolutionary Algorithms for Boundary Extraction

As we stated in Section 2, the extraction of a boundary close to the optimum through the gradient operator is harder with noisy images. For this reason, of global

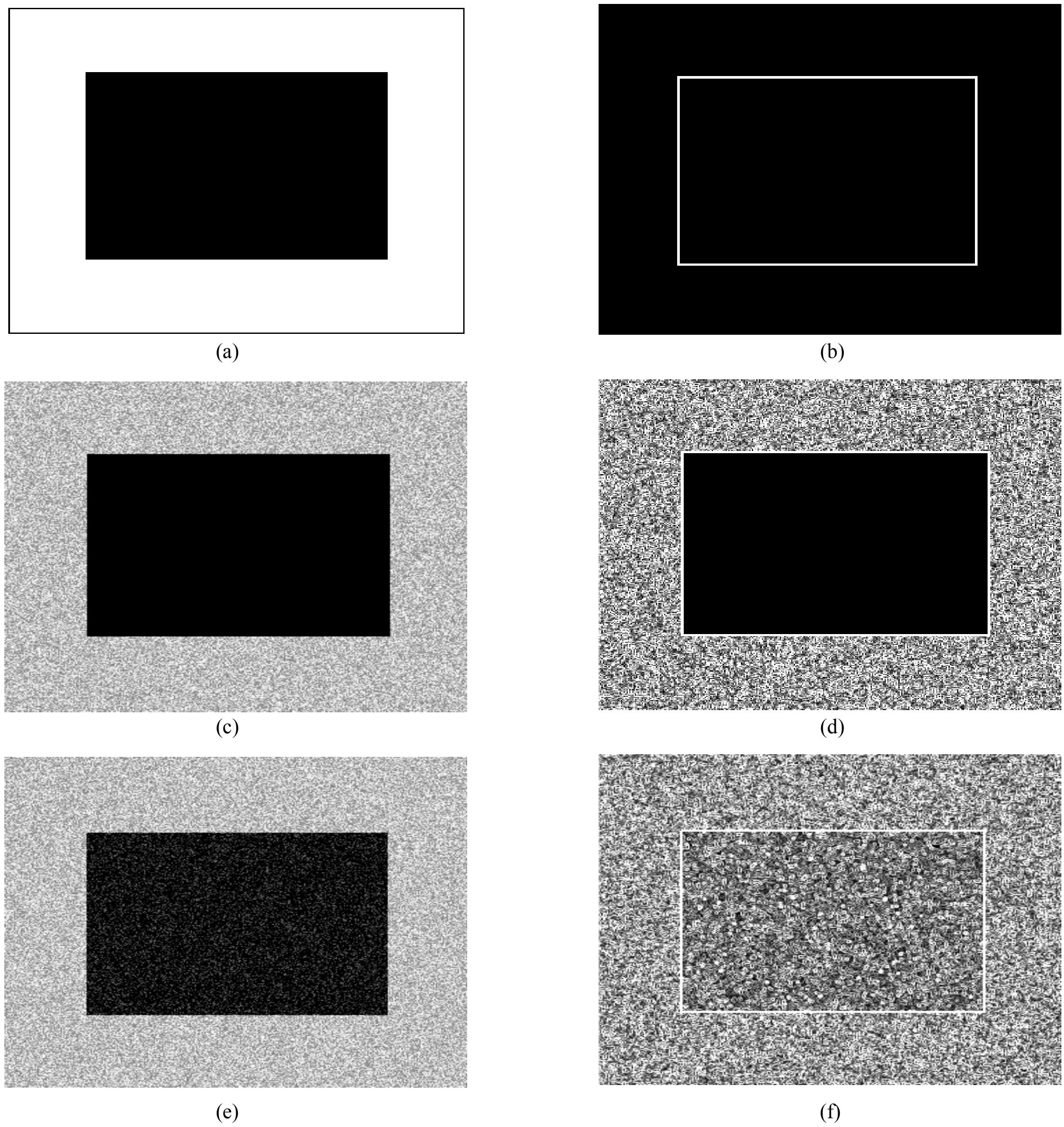

Figure 1: Input images in (a), (c) and (e) and their corresponding images of ∇ in (b), (d) and (f) respectively.

searching techniques should be employed. It is therefore sensible to regard border detection through the gradient operator not at face value, but only within an evolutionary algorithm to achieve nearly optimum boundary detection. This is in fact the idea proposed in this work. Starting from an initial population of feasible solutions (possible boundaries), the systems evolves throughout mutation and crossover operations that induce gradually better solutions. The selection is performed evaluating the fitness of the solutions via the ∇ operator.

Let's suppose the existence of a population of C_N^K contours, v. g., the ancestors. Each contour $C_i^N (i \in N)$ has a genotype c_i codified with numbers that represent the M vertices of a polygonal description of every contour. Then, a set C_N^{K+1} of offspring is created, where the genotype c_i of each contour comes form the previous genotype through a mutation that modifies only one of its vertices stochastically. Considering the image of Fig. 2(a) as a generic example, we will show in the remaining of this Section the main components of the evolutionary algorithm implemented for boundary extraction.

3.2.1 Initial Population

In any case, any possible contour that may count as an object boundary should be in the nearby of the outer limit of that given object. Consider for instance the image in Fig. 2(a), and its gradient ∇ (Fig. 2(b)). Therefore it may be sensible to start the evolutionary algorithm generating an initial population of solutions that lay close to this target, that is, a random set of contours that are in the nearby of the sought for con-

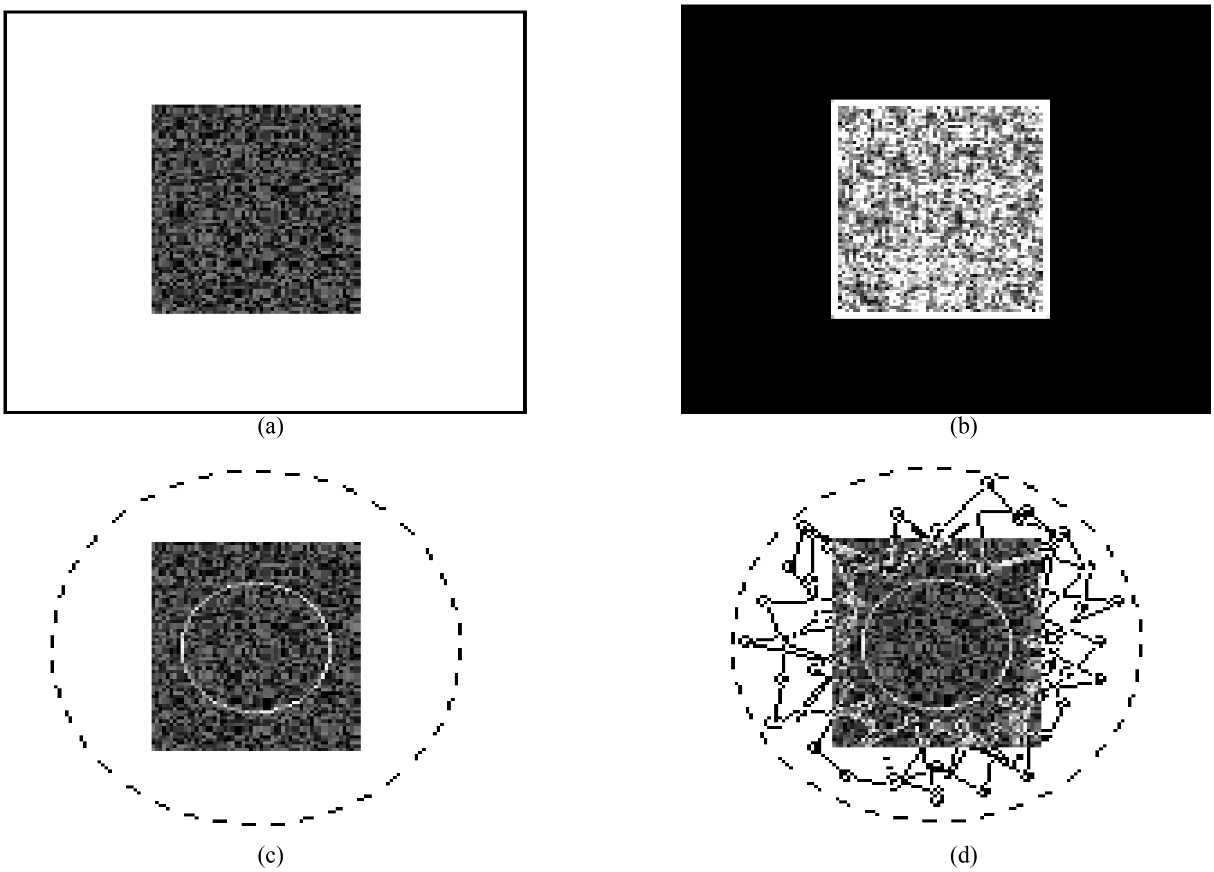

Figure 2: Generic example: input image in (a), image of ∇ in (b), circles to generate the initial population of contours in (c) and generated contours in (d).

tour. In our case this set may be created positioning two concentric circles (Fig. 2(c)), one properly including the target boundary, and other properly included in it. This could be produced in other ways, but in this work we consider this strategy of initialization only. Thus, the initial population will consist of a set of contours laying inside a ring.

Once the inner and outer radii are determined, N contours of M vertices are generated within the ring (Fig. 2(d)). Vertices are spaced at uniform angles along the ring, and the radius associated with each is randomly chosen among the minimum and maximum radii. It is important to remark that both the number of contours to generate and the angular separation of the vertices that conform these contours can be set by the user.

3.2.2 Mutation and Crossover

Given a set of contours (a generation), the next set to be considered is produced by means of a couple of genetic operators. For each of the newly generated contours, the *mutation* operator is responsible of the random selection of one of its vertices and of modifying its location. This modification is randomly produced within a radial and an angular range, being the former much larger than the later, and taking care to consider the limits of adjacent vertices. In this way a new contour is obtained modifying the original one. Fig. 3(a) and Fig. 3(b) show two contours and the correspondent offspring produced by the mutation operator.

The *crossover* operator is another widely used genetic operator. In our application, it takes the sequence of vertices of two contours, randomly chooses a place in the sequences, and swaps their vertices from this chosen place on, thus producing two completely new contours. Fig. 3(c) and Fig. 3(d) show two contours and their offspring produced by the *crossover* operator. These two operators *add* new candidate solutions, but do not *replace* their ancestors. This means that we are using a $\mu + \lambda$ evolutionary strategy, that is, previous to the selection function application, both the current population and their offspring are mixed.

3.2.3 Fitness and Selection

The evaluation of the *fitness* of a contour must quantify how close it is from the optimum. Since the optimal contour is unknown, our idea for selection is to consider fittest the “lowest cost” contour, where the cost is associated with the cumulative local intensity differences. That is, our idea of optimal contour is a set of pixels which are both linearly connected and of very similar intensity. Thus, the local cost that is accumulated at each vertex of the contour diminishes when the local gradient in the image is high. We use the following standard function to compute the local cost $k(x, y)$ in the pixel (x, y):

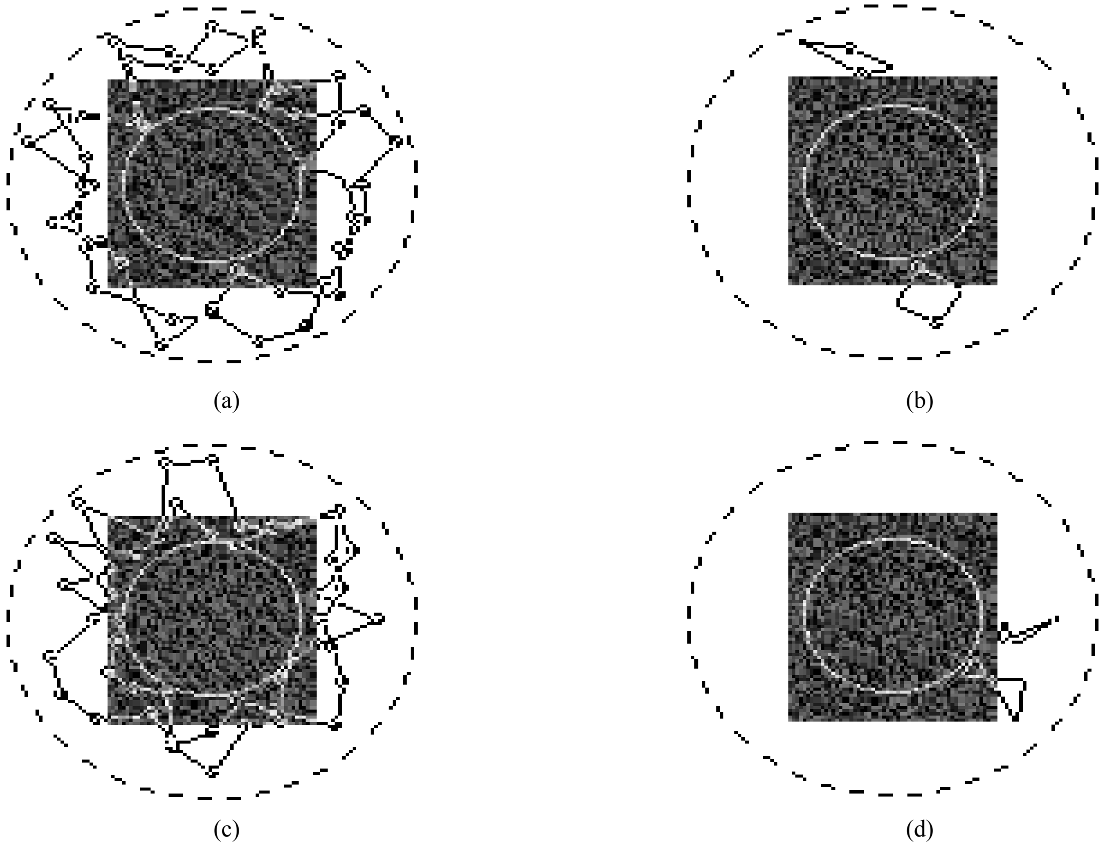

Figure 3: Mutation in (b) of the contours shown in (a), and crossover in (d) of the contours shown in (c).

$$k(x,y) = 9 - \sum_{i=-1}^{1} \sum_{j=-1}^{1} \nabla(x+1, y+j)$$

In this way the local cost at vertex c_i^j of a contour c_i^j can be expressed as $k(c_i^j)$. Masks of bigger size would imply a better estimation, but also a heavier computational cost. To evaluate the fitness $f(C_i)$ of the i-contour we only have to cumulate the local cost of each of the points that conform its genotype.

$$f(C_i) = \sum_{j=1}^{M} k(c_i^j)$$

The selection criterion is based on the elimination of those contours whose fitness is below a given relation to the maximal fitness of the present generation. That is, the population size along generations is not fixed. In practice, there is a tradeoff between this survival threshold, the computational cost, and the possibility of discarding feasible solutions. In this work we are considering the adequacy of the generated solutions, and for this reason we choose a high survival rate, thus sacrificing computational cost.

4 Expermental Results

To investigate the performance of the boundary extraction system based on evolutionary algorithms several noisy images were generated adding zero mean Gaussian noise with different standard deviations over some evaluation images.

Fig. 4 shows input images with Gaussian noise (σ=50), together with the corresponding contour found. For the object of Fig. 4(a), the number of simultaneous contours explored used was 3000 and the number of generations was 15000, while for the object in Fig. 4(d), the number of iterations was 100000 because of its greater complexity. To achieve a correct detection of larger objects (Fig. 5) the number of contours required was higher (4000) and also the number of generations (140000).

In all the cases that we tested our algorithm the results were successful, detecting nearly optimum contours in diverse conditions of noise, size and complexity of the contours to segment. For the detection of contours of larger or more complex objects, the amount of simultaneous contours and generations need to be increased. It is important to remark that all the best results were found with low mutation rates (lower than 10%).

5 Conclusions

In this work we have presented a boundary extraction methodology for digital images based on the implementation of an evolutionary algorithm. This solution features both the conceptual simplicity of the detection strategy provided by gradient operators and the robust behavior of the evolutionary mechanism. This robustness is indeed preserved under several variations in the input images, not only in the shape and

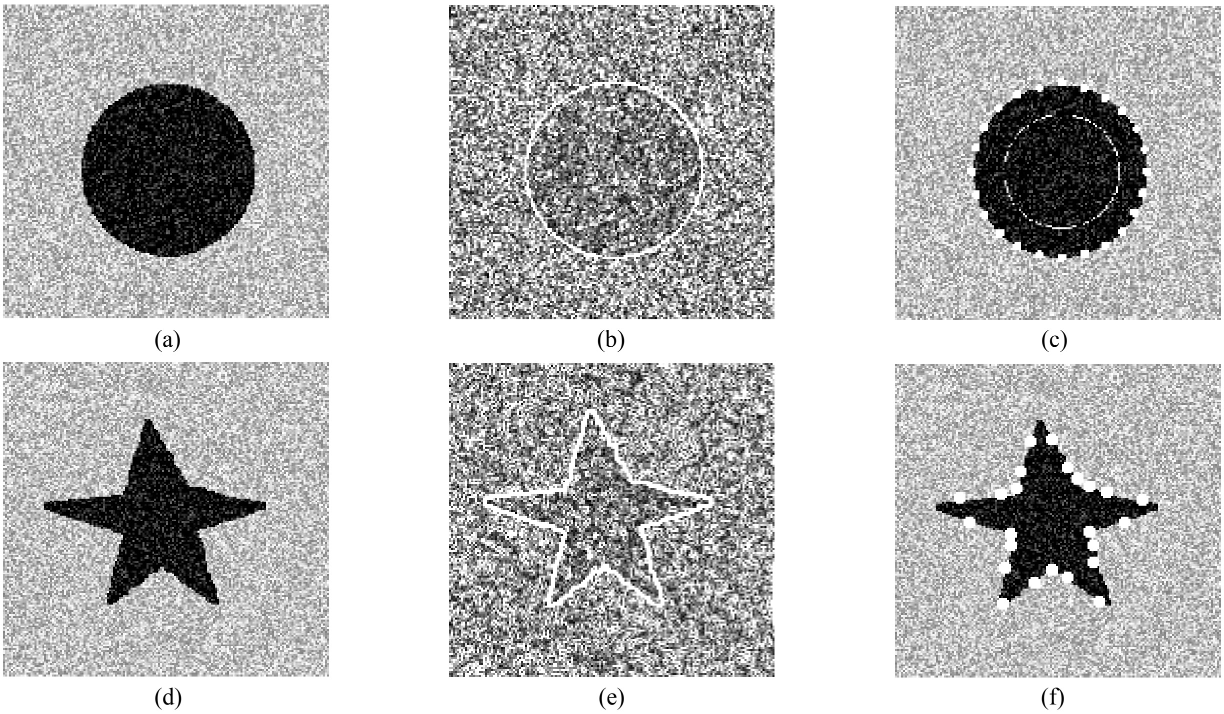

Figure 4: Evaluation images in (a) and (d), their corresponding ∇ in (b) and (e) and the detected contours respectively in (c) and (f).

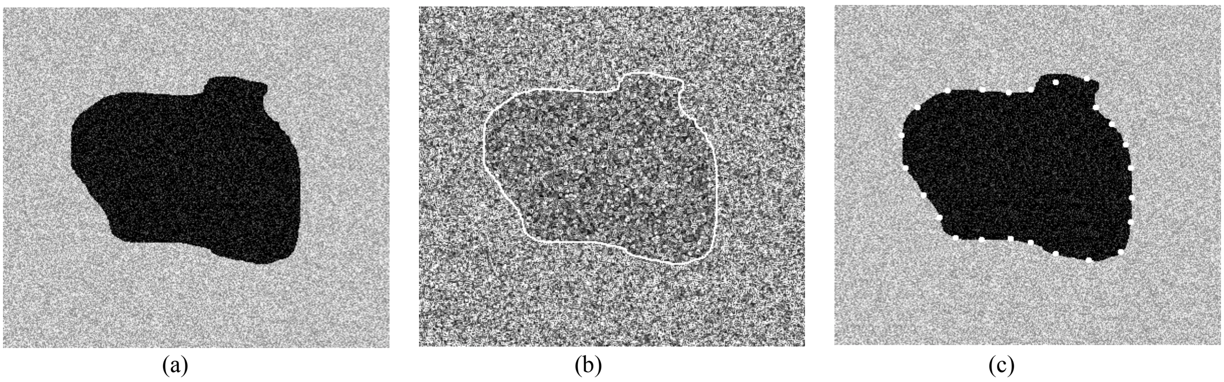

Figure 5: Larger evaluation images: (a) input image, (b) image of ∇ and (c) detected contour.

size of the objects of interest but also in difficult detection conditions, for instance non-uniform intensity levels and noise perturbations. The system performs adequately in a large evaluation set, finding adequate contours in every case in a reasonable time.

The set of genetic operators is quite standard, and can be easily extended to include the utilization and evaluation of other mutation and crossover operators, and also alternative fitness evaluation (frequency operators, transformations, etc.). These changes can be easily embedded within our working framework. Finally, the "adaptive" nature of the evolutionary system is able to support dynamic conditions in the input images for border detection "on-the-fly".

References

[1] A. Blake and M. Isard. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. Springer-Verlag, 1998.

[2] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley,1989.

[3] Rafael González and Richard Woods. *Digital Image Processing*. Addison-Wesley,1996.

[4] Anil Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall,1996.

[5] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1994.

[6] J. Park and J. M. Keller. Snakes on the Watershed. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1201–1205, 2001.

[7] D. Pham and D. Karaboga. *Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing, and Neural Networks*. Springer-Verlag, 1998.

[8] V. J. Rayward-Smith, I. H. Osman, and C. R. Reeves. *Modern Heuristic Search Methods*. Wiley, 1996.

[9] I. Rechenberg. *Evolution strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.

[10] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient Vector Flow,. *IEEE Transactions on Image Processing*, 28(3):359–369, 1998.