Enhanced Evolutionary Algorithm for Border Extraction in Noisy Images

Román Katz and Claudio Delrieux

Universidad Nacional del Sur, Dept. Electrical Eng. and Computers
(8000) Bahia Blanca - ARGENTINA
(voice) (54)(291) 4595101 ext. 3381 - (fax) (54)(291) 4595154
claudio@acm.org

Abstract. Border extraction is an important procedure associated with recognition and interpretation tasks in digital image processing and computer vision. Since local processing schemes (i.e., spatial filtering) are not appropriate for border extraction in noisy images, global and intelligent mechanisms are required to deal with this situation. Some heuristic algorithms that apply jointly local operators and some kind of optimization criterion have been successfully applied to treat images corrupted by additive noise. However, the behavior of these mechanisms is considerably undermined when managing images with multiplicative (*speckle*) noise. In this paper we present a gradient-based evolutionary algorithm that can achieve convenient boundary extraction in digital images with additive noise, and that can be successfully extended to operate with non-additive noisy images as well.

1 Introduction

Boundary extraction is an important procedure for classification and pattern recognition purposes in digital images, not only for description and interpretation tasks but also for object identification [1]. The use of gradient operators is the most common and simple tool for border segmentation. It is based on the application of a spatial filtering, i.e., the convolution of the image with a relatively small "mask" of constant coefficients (for instance, Roberts, Sobel or Prewitt). This operator detects local level variations that could correspond to boundaries between foreground and background. Although simple, gradient operators do not guarantee good border segmentation. In most situations, the results are very far from being clear-cut boundaries that surround objects, and therefore a more qualitative segmentation approach is needed.

One of the modern techniques most applied for border detection is the *active contour* approach (also called *snakes*) [2, 3]. 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 [4]. In general the active contours have limitations regarding the concavities of the contour to segment [5]. Although these methodologies may provide 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 applications. This is the case of boundary extraction in noisy images, or with non-uniform intensity level or uneven illumination.

Noise arises as a result of certain processes going on in the production and capture of the real signal. It is not part of the ideal signal and may be caused by a wide range of sources, e.g. variations in the detector sensitivity, environmental variations, the discrete nature of radiation, transmission or quantization errors, etc. The characteristics of noise depend on its source, as does the operator which best reduces its effects. However, to our purposes this is inadequate, since the image edges are more sensitive to these types of filters. Moreover, with theses filters, the noise cannot be The most common causes of noise (detection, quantization, thermal) can be modeled with an independent, additive model, allowing some assumptions which are valid for most applications. This means that each pixel in the noisy image is the sum of the true pixel value and a random, Gaussian distributed noise value. Under these conditions, a successful strategy consists on a border detection procedure by means of the gradient operator followed by some kind of global processing, for example evolutionary programming [6, 7]. This procedure was shown to work well in images corrupted by additive noise, but still present important difficulties to handle multiplicative (*speckle*) noise.

Speckle is a random, deterministic, interference pattern in an image formed with coherent radiation of a medium containing many sub-resolution scatterers. The presence of Speckle in an image reduces the resolution and the detectability of the target. Speckle noise in SAR images can be modeled as random noise that is multiplicative in nature, ie. its standard deviation is equal to its mean. Speckle appears in images as an apparently random placement of bright or dark pixels and is a direct result of SAR images being formed using radar which provides a Doppler resolution. Images with speckle noise pose severe limitations to border extraction algorithms. At the same time, SAR and ultrasound images are increasingly common, and therefore an adequate processing is required. For this reason we propose in this work a boundary extraction system that combines the use of the gradient operator within the implementation of an evolutionary algorithm that can cope with not only additive but also multiplicative noisy images. In Section 2 we present the main concepts of boundary extraction, with particular emphasis on noisy images and with non-uniform illumination conditions. In Section 3 we introduce the idea of evolutionary algorithms and their application to boundary extraction. In Sections 4 we show the results of applying evolutionary algorithms for border extraction in noisy images, and in Section 5 we present the conclusions and discuss further research lines.

2 Boundary Extraction

The most widely used feature extraction techniques, are based on a segmentation of the contours of an object by means of the detection of the boundary between foreground and background. This can be characterized as a local discontinuity in the intensity levels $I(x, y)$ with respect to neighboring pixels [1]. It is adequate then, to use a gradient operator to find these local intensity changes [8]:

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

In many occasions it is 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[1]. 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", being the last one the one used in this work. In Fig. 1 (upper right) we show the graphic of the scalar gradient ∇ of the image shown in Fig. 1 (upper left). In this image the foreground 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.

Regrettably, real world images usually contain departures from the ideal image that would be produced by our model of the imaging process. Such departures are referred to as *noise*. Noise arises as a result of unmodeled or unmodelable processes going on in the production and capture of the image. It is not part of the ideal signal and may be caused by a wide range of sources, e.g. variations in the detector sensitivity, environmental variations, the discrete nature of radiation, transmission or quantization errors, etc. Even if we acquire an image of a uniform source on an ideal camera with perfect uniformity and efficiency, the number of counts detected in all pixels of the image will not be the same. There are many kinds of noise present in an image: gamma noise, salt and pepper noise, multiplicative or negative exponential (Speckle) noise, Rayleigh noise, and uniform noise. One kind of noise which occurs in all recorded images to a certain extent is detector noise. This kind of noise is due to the discrete nature of radiation, i.e. the fact that each imaging system is recording an image by counting photons. Allowing some assumptions (which are valid for many applications) this noise can be modeled with an independent, additive model, where the noise has a zero-mean Gaussian distribution described by its standard deviation, or variance. This means that each pixel in the noisy image is the sum of the true pixel value and a random, Gaussian distributed noise value. In many cases, additive noise is evenly distributed over the frequency domain (i.e. white noise), whereas an image contains mostly low frequency information. Hence, the

[1] In the remains of this work we will refer to ∇ as the scalar gradient if not stated otherwise.

noise is dominant for high frequencies and its effects can be reduced using some kind of low-pass filter. This can be done either with a frequency filter or with a spatial filter. Often a spatial filter is preferable, as it is computationally less expensive than a frequency filter. Some types of filters can improve the signal-to-noise ratio, but the boundaries we are willing to extract are more sensitive to these types of filters. Moreover, the noise cannot be completely eliminated.

The other model for noise sources is the multiplicative or speckle noise. Speckles are an inherent characteristic of coherent imaging, including, SAR, ultrasound imaging, and other detection and sensing principles, Speckle is a random, deterministic, interference pattern in an image formed with coherent radiation of a medium containing many sub-resolution scatterers. The texture of the observed speckle pattern does not correspond to underlying structure. Speckle is a granular noise that inherently exists in all types of coherent imaging system. The presence of Speckle in an image reduces the resolution and the detectability of the target. Speckle noise in SAR images can be modeled as random noise that is multiplicative in nature, ie. its standard deviation is equal to its mean. Speckle appears in images as an apparently random placement of bright or dark pixels and is a direct result of SAR images being formed using radar which provides a Doppler resolution.

Clearly, in noisy images, border detection by means of gradient operators will fail because of the spurious fluctuations of the intensity levels, which are in fact amplified by the gradient operator above any useful threshold level. In Fig. 1 (middle and lower rows, left) we show the same image as in Fig. 1 (upper left) incorporating (respectively) additive and multiplicative noise. The resulting scalar gradient ∇ still captures the contour for a human assisted segmentation procedure (see Fig. 1 (middle and lower rows, right), but any kind of automatic segmentation is doomed to fail. 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.

If we now consider that feature extraction in general is mostly a global operation, then any successful strategy under noisy conditions should combine contour detection and global processing schemes to avoid misclassifications. This is in fact the strategy of the most known global extraction mechanisms, as the Hough transform for border detection, graph search methods [8] and dynamic programming [1]. 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 the so far segmented boundary. In this setting, a gradient operator minimizes this difference locally, but may remain stuck in local minima induced by noise. 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.

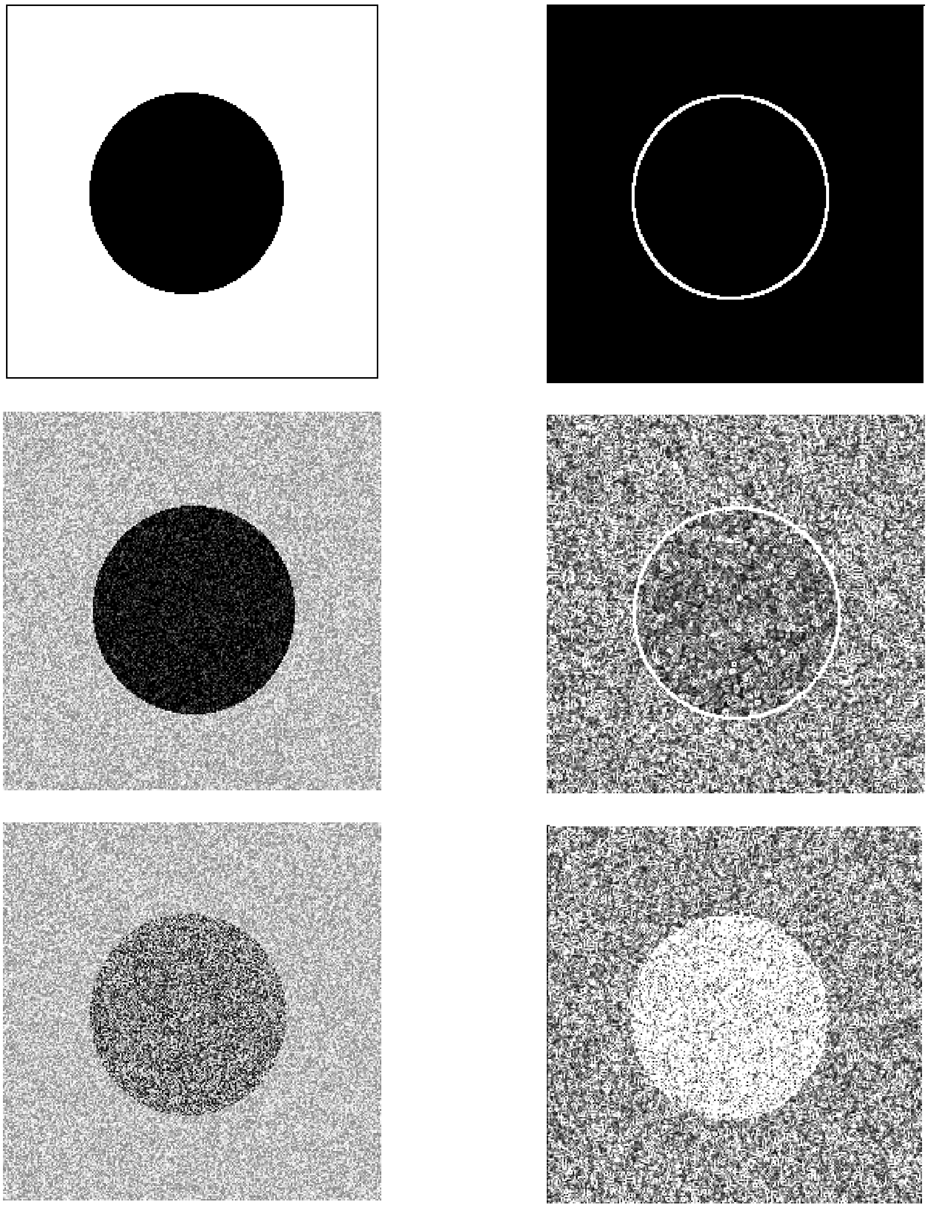


Fig. 1. A noiseless image (upper left), with additive noise (below) and with multiplicative noise (lower left), and their corresponding images of ∇ at the right of each one.

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) [10], 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 [11, 12].

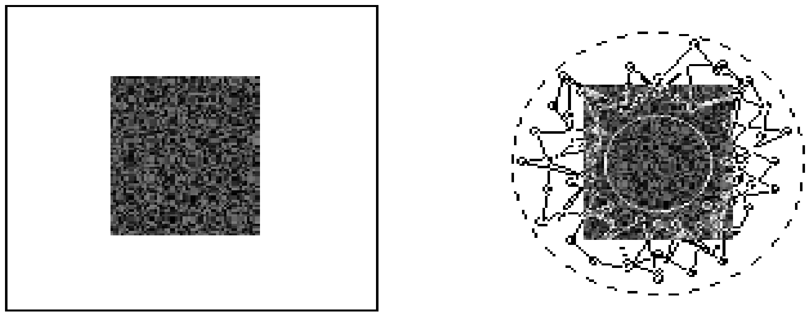
The main idea derives from a metaphor of natural evolution in biological processes, where the individuals (phenotypes) express a genetic information (genotype) and are subjected to rules of "survival of the fittest" (*fitness* and *selection*). Those best fit individuals (with higher fitness) have more 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. It is therefore sensible to regard border detection through the gradient operator within an evolutionary algorithm to achieve nearly optimum boundary detection. This is in fact the idea proposed in a previous work [7]. Starting from an initial population of possible



(a) (b)

Fig. 2. Generic example: input image in (a) and generated contours in (b).

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 $\{C_i^k\}$ contours. Here $i \in 1..N$ is an index to each of the contours in the set, and k denotes the generation we are considering. If $k = 0$ then we are considering the initial population. Each contour C_i has a genotype c_i codified with numbers that represent the M vertices of a polygonal description of every contour. Then, a set $\{C_i^{k+1}\}$ 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.

Initial Population 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). Therefore it may be convenient to start the evolutionary algorithm generating an initial population of solutions that lie close to this target, that is, a random set of contours that are in the nearby of the sought for contour (the initialization could be produced by setting the radii of a ring or by many other ways). Thus, the initial population will consist of a set of contours lying inside a ring close to the border.

Once the inner and outer radii are determined, N contours of M vertices are generated. 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.

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.

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. 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 [13].

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):

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

In this way the local cost at vertex c_i^j of a contour c_i 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 accumulate 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.

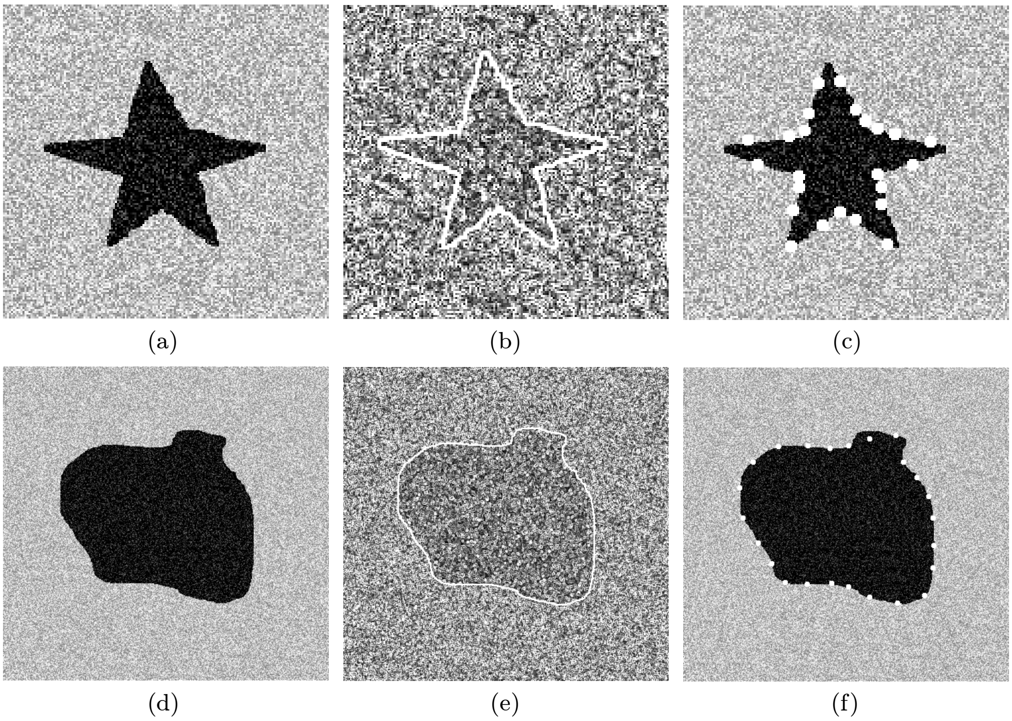
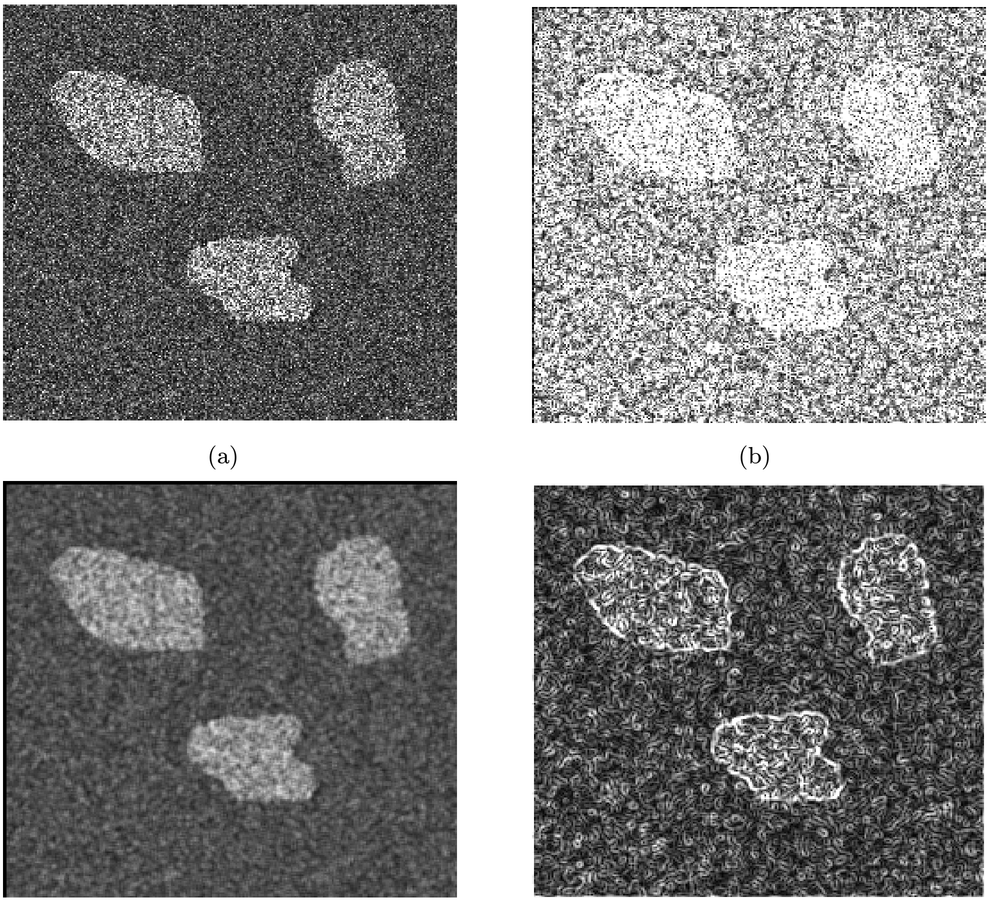


Fig. 3. Evaluation images in (a) and (d), their corresponding ∇ in (b) and (e) and the detected contours respectively in (c) and (f).

4 Experimental Results

To investigate the performance of the boundary extraction system based on evolutionary algorithms several noisy images were evaluated, for both additive noise and speckle noise. First, images with additive noise were generated adding zero mean Gaussian noise with different standard deviations over regular images. Fig. 3 shows a couple of test images with Gaussian noise (σ=50), together with their respective scalar gradient, and the final contour found with our algorithm. For the object of Fig. 3(a), the number of simultaneous contours explored used was 3000 and the number of generations was 15000, while for the object in Fig. 3(d), the number of contours was higher (4000) and also the number of generations (140000).

In all these cases the results were successful, detecting nearly optimum contours in diverse conditions of additive noise, and size and complexity of the contours to segment. The only restriction is that for the detection of contours of larger or more complex objects, the size of the population of contours and amount of generations need to be increased. For images corrupted by speckle noise, even though the algorithm worked well, it was necessary to use too many



(a) (b) (c) (d)

Fig. 4. Synthetic image with speckle noise (courtesy of Juliana Gambini) in (a), its corresponding ∇ in (b), low-pass filtered in (c) and ∇ after this filtering in (d).

initial contours and also an excessive number of iterations to achieve acceptable results. This is due to the fact that speckle noise hides almost completely the border that could be locally detected through the gradient operator ∇. Fig. 4 illustrates this issue, showing in (a) a synthetic image with speckle noise and in (b) the image of its corresponding ∇.

To improve the performance of the algorithm in cases like this, we tried several processing techniques before the application of the evolutionary algorithm is applied to the gradient of the image. Of those, the remarkable fact is that a simple low-pass filtering stage before executing the evolutionary scheme produces the best results. It produced an important improvement in the robustness and efficiency of the overall border extraction process. An explanation for this

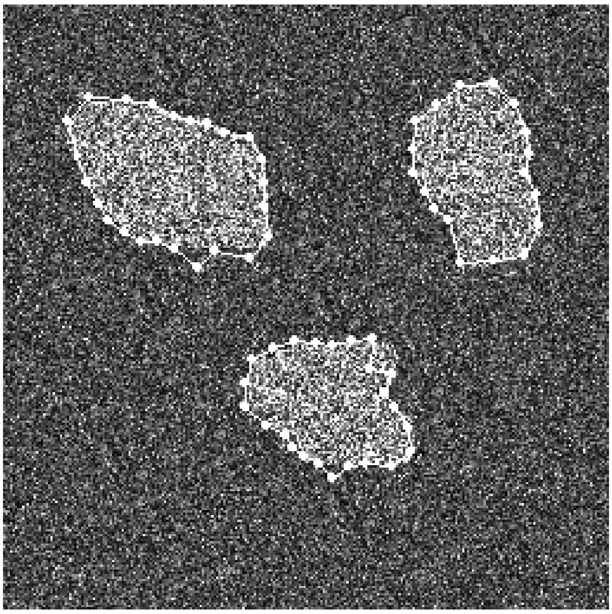


Fig. 5. The results of segmenting the borders of the Fig. 4(a) with the evolutionary algorithm applied to the image after the described filtering.

is that the low-pass of the speckle noise adds local correlation, i.e., the model is closer to an additive noise. As we stated above, low-pass filtering produces a spatial degradation in the borders. However, applying the gradient operator to the low-pass filtered image still produces a boundary that can be easily detected by the evolutionary algorithm. This is shown in Fig. 4(c) and 4(d), where a low-pass filter is applied to Fig. 4(a), and subsequently the ∇ operator. In Fig. 5. we can see the results of segmenting the borders of the Fig. 4(a) with the evolutionary algorithm applied to the image after the described filtering. In this Figure, three different initial populations were initialized around each of the brighter areas (foreground) that are present in the image.

5 Conclusions and Future Work

We have presented a boundary extraction methodology 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 size of the objects of interest but also in difficult detection conditions, for instance non-uniform intensity levels and additive noise perturbations. The system performs adequately in a large set of evaluation examples, finding adequate contours in a reasonable time.

In images with speckle noise, the algorithm needed too many initial contours and an excessive number of iterations to find good results. For this reason we explored some alternatives, and found that introducing a previous low-pass filtering stage before the evolutionary procedure produced remarkably good results in the same conditions (population size and number of iterations) that in additive noise images. This leads to the conclusion that the overall schema of filtering before boundary extraction is powerful enough to cope with even more severe interference and distortion problems. We are currently researching the use of other kind of image estimators to be applied before the evolutionary algorithm. Among these we are considering a local fractal dimension estimator, chromatic and spectral descriptors, and Hurst coefficient evaluation.

References

1. Jain, A.: Fundamentals of Digital Image Processing. Prentice-Hall, Cambridge (1996)
2. Blake, A., Isard, M.: Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion. Springer-Verlag (1998)
3. Park, J., Keller, J.M.: Snakes on the Watershed. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 1201–1205
4. Gambini, J., Jacobo, J., Mejail, M., Frery, A.: SAR image segmentation using B-Spline deformable contours. In: Congreso Argentino de Ciencias de la Computación, Buenos Aires, VIII CACiC (2002) 503–510
5. Xu, C., Prince, J.L.: Snakes, Shapes, and Gradient Vector Flow,. IEEE Transactions on Image Processing **28** (1998) 359–369
6. Katz, R., Delrieux, C.: Extracción de Contornos Mediante Algoritmos Evolutivos. In: Congreso Argentino de Ciencias de la Computación, , CACIC 2002 (2002) 588–599
7. Katz, R., Delrieux, C.: Boundary Extraction Through Gradient-Based Evolutionary Algorithm. Journal of Computer Science and Technology **3** (2003) 7–12
8. González, R., Woods, R.: Digital Image Processing. Addison-Wesley, Wilmington, USA (1996)
9. Rechenberg, I.: Evolution strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog (1973)
10. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. second edn. Springer-Verlag (1994)
11. Pham, D., Karaboga, D.: Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing, and Neural Networks. Springer-Verlag, New York (1998)
12. Rayward-Smith, V.J., Osman, I.H., Reeves, C.R.: Modern Heuristic Search Methods. Wiley, New York (1996)
13. Millado, P., Sierpe, L., Ivanissevich, L., Delrieux, C.: Testing Genetic and Evolutive Strategies in the Inverse IFS Problem. In: Proceedings of the CISST 2002 Conference, CSREA Press, ISBN 1-892512-94-7 (2002) 267–274