



**HAL**  
open science

# A Survey on Image Segmentation using Metaheuristic-based Deformable Models: State of the Art and Critical Analysis

Pablo Mesejo, Oscar Ibáñez, Oscar Córdón, Stefano Cagnoni

► **To cite this version:**

Pablo Mesejo, Oscar Ibáñez, Oscar Córdón, Stefano Cagnoni. A Survey on Image Segmentation using Metaheuristic-based Deformable Models: State of the Art and Critical Analysis. *Applied Soft Computing*, 2016, 44, pp.1-29. 10.1016/j.asoc.2016.03.004 . hal-01282678

**HAL Id: hal-01282678**

**<https://hal.science/hal-01282678v1>**

Submitted on 4 Mar 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Survey on Image Segmentation using Metaheuristic-based Deformable Models: State of the Art and Critical Analysis

Pablo Mesejo<sup>a,b,c</sup>, Óscar Ibáñez<sup>d,e,\*</sup>, Óscar Cerdón<sup>d,e</sup>, Stefano Cagnoni<sup>a</sup>

<sup>a</sup>*Dept. of Information Engineering, University of Parma, Viale G.P. Usberti 181a, 43124, Parma, Italy*

<sup>b</sup>*ISIT-UMR 6284 CNRS, Université d'Auvergne, Clermont-Ferrand, France*

<sup>c</sup>*INRIA, Univ. Grenoble Alpes, LJK, F-38000, Grenoble, France*

<sup>d</sup>*European Center for Soft Computing, 33600 Mieres, Asturias, Spain*

<sup>e</sup>*Dept. of Computer Science and Artificial Intelligence, University of Granada, C/Daniel Saucedo Aranda, 18071 Granada, Spain*

---

## Abstract

Deformable models are segmentation techniques that adapt a curve with the goal of maximizing its overlap with the actual contour of an object of interest within an image. Such a process requires that an optimization framework be defined whose most critical issues include: choosing an optimization method which exhibits robustness with respect to noisy and highly-multimodal search spaces; selecting the optimization and segmentation algorithms' parameters; choosing the representation for encoding prior knowledge on the image domain of interest; initializing the curve in a location which favors its convergence onto the boundary of the object of interest.

All these problems are extensively discussed within this manuscript, with reference to the family of global stochastic optimization techniques that are generally termed metaheuristics, and are designed to solve complex optimization and machine learning problems. In particular, we present a complete study on the application of metaheuristics to image segmentation based on deformable models. This survey studies, analyzes and contextualizes the most notable and recent works on this topic, proposing an original categorization for these hybrid approaches. It aims to serve as a reference work which proposes some guidelines for choosing and designing the most appropriate combination of deformable models and metaheuristics when facing a given segmentation problem.

After recalling the principles underlying deformable models and metaheuristics, we broadly review the different metaheuristic-based approaches to image segmentation based on deformable models, and conclude with a general discussion about methodological and design issues as well as future research and application trends.

*Keywords:* Deformable Models, Metaheuristics, Image Segmentation, Stochastic Optimization, Global

---

\*Corresponding Author. Tel. +34 666917443

*Email addresses:* pmesejo@ce.unipr.it (Pablo Mesejo), oscar.ibanez@decsai.ugr.es (Óscar Ibáñez), occordon@decsai.ugr.es (Óscar Cerdón), cagnoni@ce.unipr.it (Stefano Cagnoni)

## 1. Introduction

Image segmentation is defined as the partitioning of an image into non-overlapping regions that are homogeneous with respect to some visual feature, such as intensity or texture [1]. Segmentation algorithms are involved in virtually all computer vision systems, at least in a pre-processing stage, up to practical applications in which segmentation plays a most central role: they range from medical imaging to object detection, traffic control system and video surveillance, among many others. The importance of developing automated methods to accurately perform segmentation is obvious if one is aware about how tedious, time-consuming, subjective and error-prone manual segmentation can be.

According to the general principle on which the segmentation is based, we can build a taxonomy of the different segmentation algorithms distinguishing the following categories [2–4]: thresholding techniques (based on pixel intensity), edge-based methods (boundary localization), region-based approaches (region detection), and deformable models (shape). This paper is focused on deformable models and, in particular, on the role that stochastic optimization techniques (metaheuristics) play in their application. In fact, the segmentation problems solved by deformable models are intrinsically optimization problems. As a very general description, one can say that deformable models start from some initial boundary shape represented as a curve and iteratively modify it by applying various shrinking/expansion operations. These operations are driven by the goal of minimizing an associated energy function which, ideally, reaches its optimum when the curve perfectly fits the boundary of the object one wants to segment. Therefore, segmentation is reformulated as the global optimization of a multimodal function.

Besides the main global problem of adjusting the deformable model boundaries according to a cost function, optimization methods are at the core of different critical tasks and can be used to solve many image segmentation sub-problems, such as the selection of the parameters that regulate the algorithm (e.g., the weights of the cost function), the initialization of the curve in a location which favors its convergence onto the boundary of the object of interest, or the parameter configuration of some previous image processing step. Most often, the optimization task is solved by using classical numerical optimization methods such as Levenberg-Marquardt, Gauss-Newton or gradient descent, after relaxing the original problem such that the function being optimized becomes approximately convex [5, 6]. In addition to the selection of a proper relaxation strategy, these techniques imply that the function to be optimized is differentiable and continuous. At the same time, there is commonly a relevant probability that such local optimization methods get stuck in local minima. In fact, most local optimization techniques perform effectively when the problem under consideration satisfies the said tight math-

emational constraints. However, when the search space is non-continuous, noisy, high-dimensional, non-convex or multimodal, those methods are consistently outperformed by stochastic optimization algorithms [7, 8].

Metaheuristics are general-purpose stochastic procedures designed to solve complex optimization problems [9]. They are approximate and usually non-deterministic algorithms that guide a search process over the solution space. Unlike methods designed specifically for particular types of optimization tasks, they are general-purpose algorithms and require no particular knowledge about the problem structure other than the objective function itself, when defined, or a sampling of it (training set) when the optimization process is based only on empirical observations. Metaheuristics are characterized by their robustness and ability to exploit the information they accumulate about an initially unknown search space in order to bias the subsequent search towards useful subspaces. They provide an effective approach to manage large, complex and poorly understood search spaces where enumerative or heuristic search methods are inappropriate.

Despite their importance and the number of scientific publications on the use of metaheuristics for deformable model optimization (see section 4), this is the first manuscript that presents an overview about the area. This exhaustive survey includes all relevant papers related with the hybridization of metaheuristics and deformable models. At the same time, it aims at drawing some guidelines to help those who are willing to incorporate the advantages, and ease of use, of metaheuristics into the design of new segmentation approaches based on the same principles. The paper is structured as follows: in sections 2 and 3 we briefly review the theoretical foundations of deformable models and metaheuristics. Section 4 is the core of this survey, describing the different deformable model families into which we have divided the methods under consideration. Finally, section 5 presents some final remarks and recommendations, foreseeing, at the same time, possible future developments.

## 2. Deformable Models

The term “deformable models” was first used in the late eighties [10, 11] with reference to curves or surfaces, defined within the image domain, that are deformed under the influence of “internal” and “external” forces. The former are related with the curve’s own features while the latter refer to the image regions surrounding the curve. In other words, commonly, in deformable models, internal forces enforce regularity constraints and keep the model smooth during deformation, while external forces are defined such that the model is attracted toward an object or other features of interest within the image.

There are basically two main types of deformable models: **parametric/explicit** and **geometric/implicit**. The former represents curves and surfaces explicitly in their parametric forms during deformation, allowing direct interaction with the model and leading to a compact representation for fast real-time implementation (i.e. they are implemented by explicitly tracking control points). Alternatively, the latter can handle topological

changes naturally because these models are based on the theory of curve evolution [12–14], and they represent curves and surfaces implicitly as a level set of a higher-dimensional scalar function.

The main principles underlying both approaches are essentially similar despite the aforementioned differences and the diverse terminology found in the field: deformable models [10], deformable templates [15], active shape models [16], active contour models / deformable contours / snakes [17], deformable surfaces [18–20], active appearance models [21], or statistical shape models [22]. In the last few years, different deformable models taxonomies have been presented [15, 20, 23], according to different criteria like the mechanisms used for carrying out the contour deformation process or the geometric representations used.

Figure 1 shows the deformable models taxonomy used as reference to define the main structure of this paper. We essentially used two criteria. On the one hand, the type of deformable models representation: geometric/implicit or parametric/explicit [24]. On the other hand, the major deformable models families for which a hybridization with metaheuristics has been most frequently described in the literature. Finally, other approaches with a smaller number of contributions have also been included in section 4.5 (Brownian strings, adaptive potential active contours, Bayesian dynamic contours, and fuzzy active contours).

The next subsections will be devoted to briefly describe the main types of deformable models that have been used in combination with metaheuristics and that will be discussed in section 4.

## 2.1. Parametric Deformable Models

### 2.1.1. Active Contour Models

One of the first practical examples of parametric deformable models, called “snakes” or active contour models, was first proposed by Kass, Witkin and Terzopoulos [10, 11, 17]. An active contour model is a variational method for detecting object boundaries in images. Starting from an initial closed configuration representing a rough approximation of the shape to be segmented, an elastic model,  $C^{ini} = \{p_1^{ini}, \dots, p_n^{ini}\}$ , defined by  $n$  points, is deformed. The deformation procedure is driven by the minimization of an energy function, until the deformable model coincides with the object boundary.

Let  $X(p)$  be a parameterization of contour  $C$  and  $I$  the image intensity. Then the energy of  $C$  is given by:

$$E(C) = \alpha \int |X'(p)|^2 + \beta \int |X''(p)|^2 - \lambda \int |\nabla I(X(p))|^2 \quad (1)$$

The first two terms represent the internal energy and the third one the external energy. The internal energy is responsible for contour smoothness while the external energy is proportional to the distance between the contour and the boundary of the object to be segmented. The third term causes the active contour model to be attracted by areas with large image gradients. Since the goal is to minimize both the internal and the external energy, the external energy is defined as the negative of the gradient magnitude.  $\alpha$ ,  $\beta$  and  $\lambda$  are the system’s free

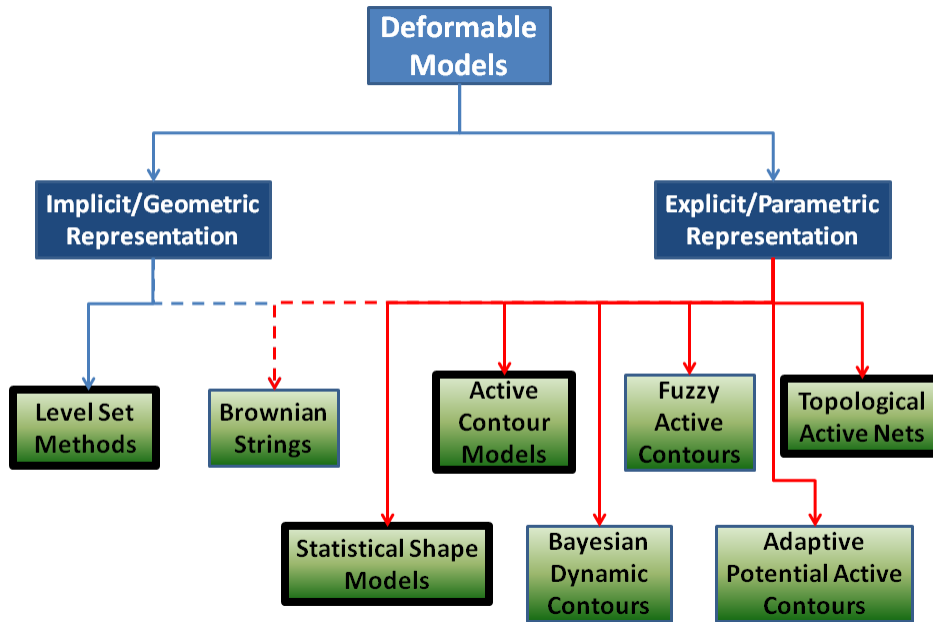


Figure 1: Deformable models taxonomy used as reference to design the main structure of this review. The “classical” and most common approaches, corresponding to the main blocks of this paper, are shown in boxes drawn in wide rims.

parameters, which are set a priori. Smaller  $\lambda$ 's reduce noise but cannot capture sharp corners, while larger  $\lambda$ 's can effectively capture irregular boundaries but are more sensitive to noise. Besides,  $\alpha$  makes the active contour model more resistant to stretching, while  $\beta$  makes it more resistant to bending. These two parameters prevent the deformable model from becoming non-continuous or to break during the iterative optimization process. The first-order derivative makes the active contour model behave as a membrane, while the second-order derivative makes the snake act like a thin plate.

Snakes represent the introduction of the general framework within which a model is matched to an image by means of energy minimization. Since external forces act in a quite intuitive manner and they can be easily adapted to track dynamic objects, these methods became very popular within the computer vision community. However, the basic method described here also presents various limitations (sensitivity to local minima, dependency on initialization, absence of prior shape knowledge, impossibility of managing topological changes) that can be addressed by alternative models described in the following subsections.

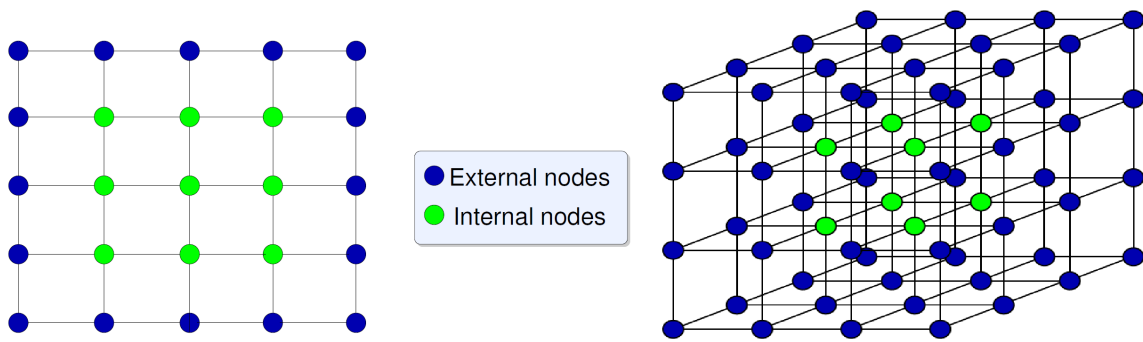


Figure 2: Topological active net and topological active volume structure representation.

### 2.1.2. Topological Active Net Models

A topological active net is a discrete implementation of an elastic mesh with interrelated nodes [25], that integrates region- and boundary-based features. The model includes two kinds of nodes: the external nodes fit the edges of the objects whereas the internal nodes model their internal topology. The main difference between topological active nets and active contour models is the capacity of the former to modify its topology: active contour models usually use only external nodes to fit the edges of the object to segment, thus lacking resources to manage topology changes. Nevertheless, the model is complex and has limitations regarding topological changes, local deformations, and the definition of the energy functional.

A topological active net is defined parametrically such that the mesh deformations are controlled by an energy function also composed by internal and external energies. The internal energy depends on the first- and second-order derivatives, which control shape contraction and bending, as well as the structure of the mesh. The external energy represents the external forces governing the adjustment process, following the two types of criteria corresponding to the internal and external nodes. As a 3-D extension of topological active nets, topological active volumes were introduced in [25, 26] and later extended in [27, 28].

In opposition to active contour models, topological active nets are able to manage topological changes that can affect the object to segment. However, this advantageous feature is penalized by the increase in the complexity of the operators needed to manage those changes.

### 2.1.3. Active Shape Models

Active shape models [29] add more prior knowledge to active contour models: they can be seen as snakes trained from examples. These shape models derive a “point distribution model” from sets of labeled points (landmarks) selected by an expert in a training set of images. In each image, a point, or set of points, is placed on the part of the object corresponding to its label. The model considers the points’ average positions and

the main modes of variation found in the training set, so the shape models are parameterized such that they represent ‘legal’ configurations. This kind of model has problems with unexpected shapes, since an instance of the model can only take into account deformations which appear in the training set. On the opposite, it is robust with respect to noise and image artifacts, like missing or damaged parts. In an active shape model, principal component analysis is commonly used to construct a point distribution model and an allowable shape domain over a set of landmark points extracted from the training shapes.

#### *2.1.4. Active Appearance Models*

Active appearance models [21] extend active shape models by considering not only the shape of the model but also other image properties, like intensity, texture or color. An appearance model can represent both the shape and the texture variability observed in a training set. Thus, it differs from an active shape model because, instead of searching locally about each model point, the model aims to minimize the difference between a new image and one synthesized by the appearance model [30]. An advantage of active appearance models with respect to active shape models is that the latter only use shape constraints and do not take advantage of all the grey-level information available within an object as the appearance models do. In turn, two of the main disadvantages of active appearance models are the difficulty of defining a training set which reliably represents the object to segment, as in active shape models, and the additional cost involved in managing a more complex model that handles more information.

#### *2.1.5. Deformable Templates*

Finally, deformable templates [15] represent shapes as deformations of a given prototype or template. Prior knowledge of an object shape is described by a, usually hand-drawn, prototype template. It consists of the object’s representative contour/edges and a set of probabilistic transformations on the template. Deformable templates have been successfully applied to object tracking [31] and object matching [32]. To define a deformable template, firstly, one needs to mathematically define a prototype which describes the prior knowledge about the object shape as the most likely appearance of the object being sought. Secondly, one needs to provide a mathematical description of the possible relationships between the template and all admissible object shapes. These descriptions represent the possible transformations which can deform the basic template and turn it into the target object as appears in the image. Then, deformable templates can be seen as a more rigid and mathematically constrained active shape model.

Geometrically deformable templates [33] are probabilistic deformable models whose degree of deformation from their equilibrium shape is measured by a penalty function associated with the mapping between the two images. A geometrically deformable template consists of a set of vertex describing the equilibrium shape (the



undeformed prototype shape), a set of vertex representing the deformed prototype shape (result of the external forces over the model), and a penalty function using a Bayesian formulation which measures the amount of deformation of the template with respect to the equilibrium shape. This penalty function is invariant to scaling, rotation, and translation of the template. Such a model can incorporate not only information about the mean shape and the variability of objects, but also information about the mean location, orientation and size of objects, as well as their variability, since it permits to segment several objects simultaneously.

## 2.2. Geometric Deformable Models

Geometric deformable models, proposed independently in [34] and [13], provide an elegant solution to address the primary limitations of parametric deformable models. These models are based on the curve evolution theory [35–37] and the level set method [14, 38]. Since the evolution of curves and surfaces do not depend on the particular way the curve has been parameterized, it is commonly considered that this modality of deformable models modifies the contour shape based only on geometric measures. In fact, when considering smooth surfaces of arbitrary topology, geometric measures are necessary because global parameterizations do not always exist. Because of this, the evolving curves can be represented implicitly as a level set of a higher-dimensional function and topological changes can be easily handled.

Amongst geometric models, the level set method for image segmentation [38] rely on an evolving closed surface defined by a moving interface which expands outwards until it reaches the desired boundary. The interface  $\Gamma(t)$  can be characterized as a Lipschitz continuous function:

$$\begin{cases} \phi(t, \mathbf{x}) > 0 & \text{for } \mathbf{x} \text{ inside } \Gamma(t) \\ \phi(t, \mathbf{x}) < 0 & \text{for } \mathbf{x} \text{ outside } \Gamma(t) \\ \phi(t, \mathbf{x}) = 0 & \text{for } \mathbf{x} \text{ on } \Gamma(t) \end{cases} \quad (2)$$

The front, or “evolving boundary”, denoted by  $\Gamma$ , is represented at time  $t$  by the zero level  $\Gamma(t) = \{\mathbf{x} | \phi(t, \mathbf{x}) = 0\}$  of a level set function  $\phi(t, \mathbf{x})$ . The evolution of  $\phi$  is commonly described by the following differential equation:

$$\frac{\partial \phi}{\partial t} + F |\nabla \phi| = 0 \quad (3)$$

known as the level set equation, where  $F$  is the speed function normal to the curve (generally dependent on time and space variables) and  $\nabla$  is the spatial gradient operator. Speed can depend on position, time, or geometry of the interface (e.g. its normal or its mean curvature). Importantly, the speed function of the level set describes the local movement of the contour and is analogue to the energy function used in parametric deformable models.

One of the main advantages of level set-based methods is the natural ability of a single surface to seemingly split apart and merge without losing its identity. On the other hand, two of their main disadvantages reside in that

they are computationally demanding and that they require considerable design effort to construct appropriate velocity functions for adapting the level set function (i.e., segmentation is the result of the choice of a suitable speed function  $F$ .)

Some hybridizations between geometric and parametric deformable models have already been presented, like geodesic active contours [39]. This approach is based on the relation between active contours and the computation of geodesics, or minimal-distance curves, connecting active contour models based on energy minimization and geometric active contours following the theory of curve evolution. The technique is based on active contours which deform in time according to intrinsic geometric measures of the image. The contours naturally split and merge during the process, making it possible to detect several objects simultaneously, as well as both interior and exterior boundaries.

### 3. Metaheuristics

Metaheuristics [9, 40] have been extensively used for continuous optimization due to a number of attractive features: no requirements for a differentiable or continuous objective function <sup>1</sup>, robust and reliable performance, global search capability, virtually no need of specific information about the problem to solve, easy implementation, and implicit parallelism. They explore effectively search spaces about which expert knowledge is scarce or difficult to encode and where traditional optimization techniques fail. Metaheuristics are non-deterministic and approximate algorithms, i.e., they do not always guarantee they can find the optimal solution, but a good approximation in reasonable time. They are not problem-specific, permitting an abstract description level, even if they may make use of domain-specific knowledge to enhance the search process.

The main objective of these optimization/learning procedures is to achieve a trade-off between intensification and diversification. Diversification (exploration) implies generating diverse solutions to explore the search space on a global scale while intensification (exploitation) implies focusing the search onto a local region where good solutions have been found.

Metaheuristics can be taxonomically divided into:

- *Trajectory methods*. The search process describes a trajectory in the search space and can be seen as the evolution in (discrete) time of a discrete dynamical system. Examples of this category are tabu search [41], simulated annealing [42, 43], iterated local search [44] or variable neighborhood search [45].
- *Population-based methods*. These techniques deal, in every iteration of the algorithm, with a population

---

<sup>1</sup>In this paper we refer to the objective function as 'fitness function', a term which would rigorously be appropriate only for genetic algorithms, but which is most frequently adopted when referring to other metaheuristics, with particular regard to bio-inspired ones.

of solutions. In this case, the search process can be seen as the evolution in (discrete) time of a set of points in the solution space. Paradigmatic cases in this regard are evolutionary algorithms, such as genetic algorithms [46–50], evolution strategies [51, 52] (within which it is worth mentioning the recent Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [53]), and evolutionary programming [54], and swarm intelligence techniques [55], such as particle swarm optimization [56]. Evolutionary algorithms are based on a computational paradigm which replicates mechanisms inspired by those of biological evolution, such as reproduction, mutation, recombination, and selection, to solve optimization problems. Swarm intelligence methods emulate the collective behavior of decentralized, self-organized artificial systems, in which global search is an emergent behavior of a population of “agents”, which are singularly programmed to perform a same, much simpler task. Differential evolution [57–60] is a more recent and increasingly popular algorithm for continuous optimization which inherits its features from both evolutionary algorithms and swarm intelligence methods.

- *Memetic algorithms.* These techniques [61] are hybrid global/local search methods in which a local improvement procedure is incorporated into a population-based algorithm. The idea is to imitate the effect of learning and social interaction during the life span of an individual by some kind of (local) improvement mechanisms applied to the solutions found by the usual global search operators. While this general definition allows memetic algorithms to include a virtually infinite variety of possible hybridizations of existing methods, some of these algorithms originated independently of other previously existing ones and constitute a specific niche within this class. Among these we would like to mention scatter search [62].

Within each of these classes, a huge number of other methods exist that we have not listed, as we decided to mention here only those which will be most frequently considered in the approaches to metaheuristics-based deformable models described in the next section.

A detailed description of metaheuristics is not within the scope of this paper, nor relevant for most of the usual readers of this journal. Researchers in computer vision and all others who are not familiar with these methods may refer to the papers cited in the previous list or to one of the many excellent textbooks which exist on the topic, such as the ones by Eiben and Smith, or the one by Engelbrecht [63, 64].

Despite the impressive empirical evidence of their effectiveness, metaheuristics have been criticized for not being perfectly defined as a unified and formal paradigm. In many cases, their performance on classes of functions where other methods fail have more than balanced those criticisms [65]. In addition to this, some relevant mathematical proofs can be found in the literature that study the convergence properties of these algorithms and demonstrate their power. For instance, Holland’s schema theorem is a quantitative explanation of the optimization ability of binary-coded genetic algorithms [46]. In terms of asymptotic convergence, many studies show

the convergence properties of evolutionary algorithms in which the algorithms are generally limited to mutation operators and elitist replacement [66]. More theory on convergence exists for evolutionary strategies [67] and swarm intelligence algorithms [68, 69]. A very rigorous theoretical study of genetic algorithms can be found in [70], which deals mainly with Markov chains and dynamical systems approaches. Moreover, in [71], an overview of the basic techniques for proving convergence of metaheuristics to optimal solutions is given. In general, a good combination of the major components of such algorithms (intensification/exploitation and diversification/exploration) usually ensures that global optimality is achievable [63, 72].

#### 4. Image Segmentation using Deformable Models and Metaheuristics

To understand the growing importance of this research field, the histogram in Figure 3 displays the evolution of the scientific production in metaheuristics-based image segmentation. The graph has been obtained by executing the Elsevier Scopus™ query:

```
"image" AND ("segmentation" OR
"localization" OR "partition")
AND ("differential evolution" OR "swarm"
OR "chc" OR "evolutionary"
OR "genetic algorithm" OR "cma-es"
OR "PSO" OR "annealing" OR "tabu search"
OR "metaheuristic")
```

In order to know the volume of publications on image segmentation using metaheuristic-based deformable models, the same query was executed adding a reference to terms related to deformable model (*AND (deformable model OR template OR topological OR statistical shape OR active models OR active nets)*). The result included 165 publications and an h-index of 16 related to this topic since 1995 (see Figure 4). It is clear that this work has had a great impact on the scientific community, as shown by the large number of citations. On the other hand, considering the relevant properties of metaheuristics, a more frequent application to image segmentation problems seems inevitable and, even more, to segmentation using deformable models. Given the great potential of these two families of methods (metaheuristics and deformable models), it seems a safe bet to explore the possibilities offered by their combination by studying new methods or improving the existing ones, making algorithmic implementations more efficient, and introducing novel approaches to image analysis fully based on metaheuristics or for which metaheuristics are an essential part [73].

In the literature, only two surveys can be found, which are just partially related to this one: [74] and [8]. The former provides a state-of-the-art survey on the application of the principles of genetic algorithms to medical

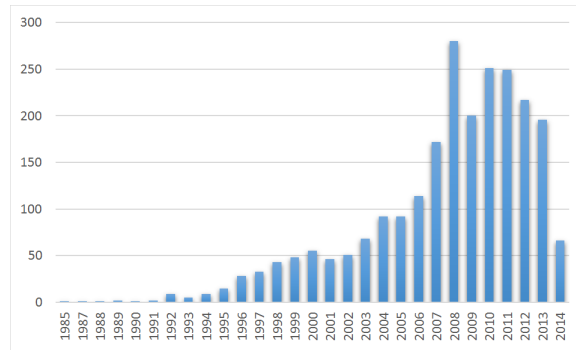


Figure 3: Scientific production related to image segmentation using metaheuristics. The queries, run in Elsevier Scopus<sup>TM</sup> on the 25th of June, 2014, displayed 2347 papers with a globally growing tendency.

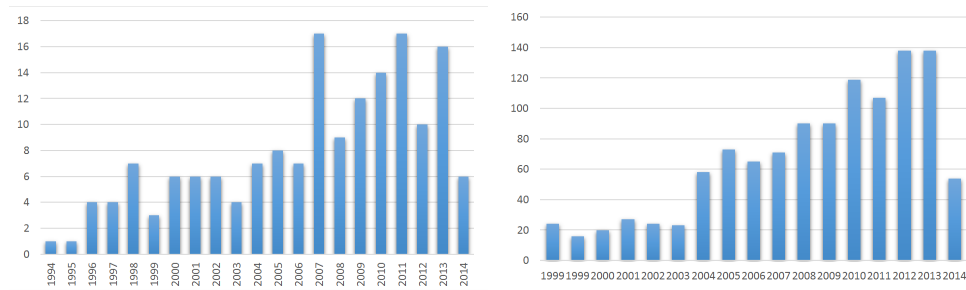


Figure 4: Publications (left) and citations (right) related to image segmentation using deformable models and metaheuristics. The queries, run in Elsevier Scopus<sup>TM</sup> on the 25th of June, 2014, displayed 165 papers and 1137 citations with a clear growing tendency.

image segmentation, only focusing onto this metaheuristic and considering all kinds of segmentation techniques. In the latter, the author examines the suitability of a number of different optimization methods (differential evolution, genetic algorithms, self-organizing migrating algorithm, simplex, and pattern search) for the task of template matching, performing tests over a series of 2-D analytic functions designed to highlight the generic properties of each optimization method, as well as on three images of increasing difficulty.

The next sections review the main metaheuristic approaches applied to deformable model optimization. While every section shows the same structure to provide a consistent description, each of them is devoted to a specific deformable model type, analyzing the problems it is able to solve and the difficulties which have to be tackled in applying such models to real-world problems.

#### *4.1. Active Contour Models*

Among all kinds of deformable models, the active contour model is the one which has been most frequently hybridized with metaheuristics as testified by the large number of works, the different roles played by the metaheuristics, and the diverse application fields. The earliest proposals date from the late nineties [75–77] with an important number of contributions already in those first years. Since those initial pioneering results, genetic algorithms have been the type of metaheuristic which has been used most frequently. In recent years, swarm intelligence has also attracted the attention of researchers in classic contour models which resulted in several publications [78–86]. Finally, a few proposals have been based on simulated annealing [76, 87–90] and basic memetic approaches [90–92]. Applications of more recent metaheuristics such as differential evolution, CMA-ES, or scatter search are absent, up to now.

Metaheuristics play different roles in the existing approaches to image segmentation based on active contour models. Most methods employ a metaheuristic to evolve each control point of the model, searching for its best location in a given search space. Many of these approaches are extensions or applications to different segmentation problems of the pioneering work by Ballerini [75], which introduced for the first time the so-called “genetic snakes”. Another important set of proposals follow a coarse-to-fine approach employing different strategies: multi-scale segmentation [93, 94], multi-stage evolution [77, 90, 95, 96], or the combination of different optimization methods at different stages [90, 92, 97].

A completely different approach is followed in [91, 98–100] where the specific metaheuristic is used for searching the best set of values for the parameters that control the evolution of the active contour model, i.e., the weights of the different terms of the energy function. All these works propose a training mechanism to obtain the best value for those parameters using manually segmented examples as training data for the supervised learning process. Similarly, in [101], a supervised approach delivers a global set of parameter values as in the aforementioned proposals, while an unsupervised approach also determines a local set of parameter values.

Other methods take advantage of both the learning and optimization capabilities of genetic algorithms in such a way that the active contour model is evolved jointly with the energy terms weights [102].

Finally, there are few approaches where the metaheuristic plays an alternative role. In [81, 99, 103] metaheuristics are used to set the initial location of the snake, the number of required snakes, and the appropriate number of control points. Far from other approaches, but still using a metaheuristic inside the deformable model-based image segmentation process, the authors in [89] use simulated annealing to set the state of each neuron of a Hopfield neural network which optimizes the energy function.

#### 4.1.1. Encoding

The existing approaches within this category can be classified in metaheuristics which: evolve control points, learn energy term weights, and initialize the model. Each of them considers a different encoding scheme.

Among the proposals that employ a metaheuristic to evolve the control points of an active contour model, most of them encode the positions of the snake in the image plane in polar coordinates. The total number of snake control points is stored in the chromosomes. This is the encoding scheme employed in all the works developed by Ballerini [75, 106, 108, 116, 117] and other related approaches [119, 122]. Alternatively, other works encode the location of each control point using Cartesian coordinates. Some of these employ Gray-coding [93, 94, 96, 113, 114] while others use real-number encoding [92, 97].

In the same group of proposals, there are few methods that do not encode the control points coordinates. In [77], each gene of the binary-coded chromosome represents the state  $S^i$  of the contour  $C^i$  according to a number of parameters (control point and associated search space for each of them)  $V_j^i$ , where  $S^i$  is defined as the contour state after the  $i^{th}$  state transition. These values correspond to displacement vectors in a state map, which are added to  $V$  to give the variant contour state. In [118], for every control point the authors encode the distance and the angle from the center to the control point. In [95] the chromosome consists of three 2-D matrices whose elements are the Fourier descriptors [124], and in [110, 111] the chromosome is an integer vector containing the index value (location in a neighborhood window within which individual control points can move) for each of the 4 control points, representing the location they moved to. Approaches based on swarm intelligence also encode similar data. In [82] and [86], each particle's search space is constructed around each control point while in [79, 83, 84] each control point in the snake is associated to a swarm of particles searching a large window which surrounds it.

In the second group of proposals, where the energy term weights are learned using metaheuristics, the information encoded by each method differs from one to another even if they all employ a genetic algorithm. In [99], two different parameter sets are learned from examples: the edge detector (filter coefficients, offsets of the data to be processed, and a threshold) and the deformable model parameters (the weights of an uncertainty

Table 1: Active Contour Models (or Snakes). This table continues in Table 2. From left to right: author(s) and reference to the paper, deformable model type, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Name	Dim.	Metaheuristic	Image Modality	Year
Kidder and Nnaji [76]	Discrete surface	3-D	Simulated Annealing	not specified	1997
MacEachern and Manku [77]	Snake	2-D	Genetic Algorithms	not specified	1998
Cagnoni et al. [99]	Snake	3-D	Genetic Algorithms	3 synthetic seqs. of 64 imgs.: 1 seq. of computed tomography (right lung) 2 seqs. brain magnetic resonance imaging	1999
Tanatipanond and Covavisaruch [94]	Snake	2-D	Genetic Algorithms	60 Brain magnetic resonance images	1999
Tanatipanond and Covavisaruch [93, 104]	Snake	2-D	Genetic Algorithms	60 Brain magnetic resonance images	1999
Ballerini [75, 105]	Snake	2-D	Genetic Algorithms	12 synthetic and 2 ocular fundus images	1998-1999
Ballerini [106, 107]	Snake	2-D	Genetic Algorithms	unspecified number of dynamic angiographic images	1999
Ballerini [108, 109]	Snake	2-D	Genetic Algorithms	13 synthetic, 2 retinal, and 1 photo (meat)	2001-2007
Bredno et al. [98]	Balloon Snake	2-D	Genetic Algorithms	137 histochemically stained cells and unknown number of endoscopic seq. of larynx	2000
Chen and Sun [91]	Snake	2-D	Taguchi's method + Genetic Algorithms	1 synthetic and 12 cardiac ultrasonic images	2000
Liatsis and Ooi [110, 111]	Snake	2-D	Coevolutionary Genetic Algorithms	road scenarios unspecified number of real road images (car and trucks)	2001-2002
Tohka [92]	Simplex Mesh	3-D	Memetic Algorithms	9 synthetic surfaces	2001
Fan et al. [95]	Dynamic Surface	3-D	Parallel Genetic Algorithms	unspecified number of brain MRIs	2002
Xu et al. [112]	Snake	2-D	Genetic Algorithms	unspecified number of mammographies	2002
Seguier and Cladel [113, 114]	Snake	2-D	Genetic Algorithms	1 video sequence (lips)	2003
Ballerini and Bocchi [115, 116]	Snake	2-D	Genetic Algorithms	unspecified number of hand X-Ray	2003
Ballerini [117]	Snake	2-D	Genetic Algorithms	unspecified number of video seq. (people)	2003
Mishra et al. [96]	Snake	2-D	Genetic Algorithms	cardiac ultrasound seq. (20 frames and 10 patients)	2003
Rousselle et al. [101]	Snake	2-D	Genetic Algorithms	unspecified number of synthetic images	2003
Plissiti et al. [89]	Snake	2-D	Neural Network + Simulated Annealing	18 arterial segments of 1432 ultrasound frames	2004
Mun et al. [118]	Snake	2-D	Genetic Algorithms	6 synthetic, 2 brain magnetic resonance imgs 2 microscopy and 2 photos	2004



Table 2: Active Contour Models (or Snakes). This is the second part of Table 1. From left to right: author(s) and reference to the paper, deformable model type, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Name	Dim.	Metaheuristic	Image Modality	Year
Wang et al. [86]	Snake	2-D	Ant Colony Optimization	unspecified left ventricle ultrasound images	2005
Mobahi et al. [81]	Snake	2-D	Swarm Intelligence	20 imgs. of faces/hands/lips	2006
Wang et al. [85]	Snake	2-D	Particle Swarm Optimization	not specified	2006
Rad and Kashanian [119]	cubic B-spline	2-D	Genetic Algorithms	1 mammography	2006
Tseng et al. and Li et al. [79, 83, 84]	Snake	2-D	Particle Swarm Optimization	3 synthetic, 1 computed tomography, 1 ultrasound	2006-2009
Delu et al. [78]	Snake	2-D	Particle Swarm Optimization	not specified	2007
Wu et al. [120]	B-spline	2-D	Genetic Algorithms	Synthetic images and 2 tooth and mandible computed tomography seq.	2007
Hong and You-ru [121]	Snake	2-D	immune Genetic Algorithms	2 synthetic images	2007
Tang et al. [90]	Snake	2-D	Simulated Annealing	1 synthetic and 1 magnetic resonance images	2007
Cristhian et al. [87, 88]	Snake	2-D	Simulated Annealing	90 X-ray tomography of knots of wood samples	2008-2012
Hussain [122]	Snake	2-D	Genetic Algorithms	unspecified number of retinal images	2008
Teixeira et al. [100]	Gradient Vector Flow Snake	2-D	Genetic Algorithms	150 Cardiac magnetic resonance images	2008
Hsu et al. [103]	Poisson Gradient Vector Flow	2-D	Genetic Algorithms	3 × 16 liver positron emission tomography slices	2008
Vera et al. [97]	Snake	2-D	real-coded Genetic Algorithms	6 angiographic seq. of the left ventricle (228 images)	2010
Shahamatnia and Ebadzadeh [82]	Snake	2-D	Particle Swarm Optimization	5 Synthetic, 1 computed tomography, and 2 Brain magnetic resonance imgs.	2011
Talebi and Ayatollahi [102]	Snake (Balloon)	2-D	Genetic Algorithms	unspecified number of breast ultrasounds	2011
Chen et al. [123]	Snake	2-D	Genetic Algorithms	452 samples of handwritten Chinese characters	2012

function used to compute the elastic contour model coefficients). All sixteen parameters have been encoded as 6-bit binary strings (resulting in a 96-bit genome), even if all parameters are real values. In [98], the parameters of a Balloon model are encoded. In particular: (i) the minimal and maximal edge length, (ii) the scale of gradients, (iii) the appearance and intensity of gradients, (iv) the mapping of image values to image potentials, (v) the strength of the deformation force, and (vi) the sign of the pressure. Each individual  $i$  in this population is a parameter set  $P_i = \vec{p}_{i,j}$  which consists of a parameter's value  $val_{i,j}$  combined with its variability during reproduction  $var_{i,j}$  (that helps to increase or decrease the diversity of an evolving population). A different set of parameters is encoded in [100] for a Balloon model as well. The set of parameters are: elasticity; rigidity; viscosity; the largest amplitude associated to the image gradient; and  $k_1$  and  $k_2$ , parameters that weight the strength of the Balloon forces. All of them are binary-encoded. In contrast to the former three approaches, Chen and Sun [91] learn local weights, different for each control point, instead of global ones. Thus, they encode the corresponding weights  $\alpha_{ij}$  for each control point  $P_i$  and each feature  $j$ . A combination of global and local learning is employed by Rousselle et al. [101]. Their genetic algorithm comprises two different phases which rely on a similar encoding. In the first phase, a genetic algorithm learns the (global) snake parameters by evolving individuals who encode them (weights of the five energy function terms: continuity, curvature, gradient, intensity, and balloon force). In the second, the same parameters are encoded for each point of the snake. Finally, in [102], each of the four control coefficients of a Balloon model (continuity, curvature, image, pressure) are encoded as a set of ten binary genes.

The last group of methods use metaheuristics as part of the initialization mechanism of the active contour model. Two approaches fit in this category. In [103], the threshold and sigma parameters of the Canny edge detector are encoded into a binary chromosome. The other method [81] uses swarm intelligence for snake initialization. The behavior of each agent is defined by three state variables, namely position, velocity, and energy.

#### 4.1.2. Operators

It is of common knowledge in evolutionary algorithms that the implementation of most of the operators depend on the encoding scheme employed, mainly, binary or real-coded. *Ad hoc* operator design could be needed to overcome specific restrictions such as the prohibition of link crossings in snake models. However, restrictions can be neglected in the definition of the operators if they are expressed as penalty terms in the fitness function [125]. The latter is the most common approach followed by the different snake-based methods which tend to employ standard operators and penalize unfeasible solutions through the fitness function.

This way, most of these approaches use uniform mutation (bit-flip in the case of binary encoding). A few proposals do not use the mutation operator at all [92, 95, 98] or use something different from the uniform

scheme. Among the latter, [97] uses a non-uniform mutation; [110, 111] use the adaptive non-uniform mutation proposed by Michalewicz [48]; and [77] is the only proposal with an *ad hoc* mutation design which randomly reassigns control-point states in population members to alternate states. The mutation rate is related to the number of bits used to completely encode the chromosome.

With respect to the selection mechanism, roulette wheel is the most frequently employed method [75, 98, 100, 106, 108, 116, 117, 119, 122], followed by tournament selection [92, 97, 101]. A rank-based selection operator was implemented in [77] while in [110, 111] the authors differentiate between inter- and intra-population selection mechanisms. In the intra-population case three chromosomes are selected and the two with the largest average distance between control points are chosen as parents to favor diversity (a kind of family competition approach [126]). In inter-population selection, populations that share control points with the population of interest (*reference populations*) are termed *candidate populations*. Chromosomes from the reference and candidate populations are known as the major and minor chromosomes or parents, as appropriate. According to this scheme, a minor parent is selected from a randomly selected candidate population, for being crossed with a major parent. Elitism is only employed in a few proposals [97, 100].

Most proposals use standard crossover operators. In particular, all the binary-coded approaches use two-point [75, 101, 106, 108, 116, 117, 119, 122], one-point [91, 99], or uniform crossover [77, 102]. However, for real-coded chromosomes there is no standard approach and several *ad hoc* crossover operators have been proposed. Among standard real-coded crossover operators, BLX- $\alpha$  is employed in [92, 100] and linear crossover in [97]. Concerning ad hoc proposals: two different crossover operators are implemented in [98], each of which is applied to a fixed number of individuals. The former only considers a single parent that creates one child, whose parameters are randomly set with a uniform distribution of the parameters in the range  $[val_{i,j} - \frac{1}{2}var_{i,j}, val_{i,j} + \frac{1}{2}var_{i,j}]$ . The latter crossover first sets the mean of all parameters values (from both parents) and then changes them according to the range defined in the single-parent crossover. In [110, 111] the authors propose a coevolutionary genetic algorithm [127], where intra- and inter-population crossover operators are employed, as in the case of the selection operator. The former uses uniform, arithmetic and linear arithmetic crossover operators, while in the latter a “major” parent is split at a randomly chosen location, such that a valid child can be generated by substituting one of its ends with an appropriate segment from the “minor” parent.

Finally, the way the population is initialized is much more variable. In [75, 106, 108, 116, 117, 119], the initial population is randomly originated within a region between two user-defined radii  $r$  and  $R$ . Hussain et al. propose an approach [122] that only differs from Ballerini’s in the initial population and in the setup of the neighborhood, since they use a radial instead of a square window. In [98] the genetic algorithm starts by sorting the individuals according to their fitness function and selecting a fixed number of single-ancestor reproductions

(this random selection chooses parents with a Poisson-distribution so that individuals with higher fitness are more likely to become a parent). The initial variability  $var_{i,j}$  of all parameters is set to one fifth of their absolute initial value  $val_{i,j}$ . In [97], the initial population is generated by randomly modifying the landmarks detected within a given neighborhood and, in [91], the weights obtained by Taguchi’s method are used as the genetic algorithm’s initial population.

Besides evolutionary algorithms, there are a few proposals based on swarm intelligence which do not use the operators described above. The method described in [82] is a customized particle swarm optimization algorithm that avoids concave boundaries and local minima, while limiting sensitivity to noise. It considers the particles’ average position at time  $t$ , which approximates the center of mass of particles, to speed up the algorithm and solve problems when the snake stagnates and there is no other compelling force. It also includes an external energy term in the particle swarm optimization equations. In [79, 83, 84] the initialization and subsequent movements of the particles in the swarm of particles  $O_i$  is constrained within a certain search window (in particular around the perpendicular bisector of the line connecting two neighboring control points). This method is intended to avoid dramatic changes which may cause improper snake evolution such as the crossing of the snake with itself.

Finally, a representative of simulated annealing-based approaches is proposed in [87, 88]. A classical simulated annealing algorithm is used. A Boltzmann distribution [43] is assumed and, consequently, a logarithmic cooling schedule is used. This method initially considers the generation of a contour in the 2-D image space having a given number of vertices. A neighborhood patch of  $n \times n$  pixels is created for every pixel in the contour. Then, in that neighborhood, a new candidate vertex is selected by evaluating it according to the energy function. The acceptance of the new vertex depends on criteria such as the number of iterations, number of acceptances and the probability of accepting a “bad solution”. Additionally, a criterion for the elimination and creation of new vertices is used. This criterion is based on the distance between the vertices in the contour. Once all vertices in the contour have been checked for a given temperature  $T$ , the criterion for elimination and creation is applied. A vertex is eliminated in case the distance to its neighbor is smaller than a minimum distance ( $d_{min}$ ). Similarly, a vertex is created in case the distance between two vertices is higher than a maximum allowed distance ( $d_{max}$ ).

#### 4.1.3. *Fitness Function*

In all approaches employing a metaheuristic to evolve the control points of an active contour model, the fitness function to be minimized is the total snake energy. However, the active contour model terms change from one method to another depending on the object(s) to be segmented. In fact, the external terms in [75, 106] include the gradient of the image plus a slightly different edge functional whose minima lies on the zero-crossings of  $\nabla^2 G_\sigma * I(x,y)$ . These methods also include an energy term associated to the eye’s foveal avascular

zone, which is the object to be segmented in this application. That leads to a modified version of the image energy, which considers both the magnitude and the direction of the gradient and the Laplacian of Gaussian. The same formulation employed in [106] is extended in [119] by defining the contour as a piecewise parametric curve using a cubic B-spline. The authors adapt Lai and Chin [128] curvature and continuity terms to the cubic B-spline model. In [108] the author extends the formulation of genetic snakes [75] to segment connective tissue in meat images, exploring additional internal and external energy terms and applying them to color images. This is done by introducing a new energy term involving the gradient of the RGB color components, which causes the contour to expand or contract depending on the sign of a given constant  $\delta$ . An area energy term is also introduced which forces the snake to enclose a given reference area  $A_{ref}$ . This term has the form of a harmonic potential reaching a minimum when the area enclosed by the snake is equal to the reference area. In [115, 116] a priori knowledge is incorporated into the fitness function in order to segment bones in radiographic images. This is obtained by using three snakes representing the three bones to segment, chained together by means of the binding force. Information about the geometry of the bones is not necessary but only their relative position. Each snake is composed of 36 points and the binding energy acts on five pairs of consecutive points in each junction. Apart from the previously proposed internal and external energy terms, other two terms have been added: i) a derivative energy that is minimal when the snake is positioned on the image edge, having the brighter region to the left of the snake and the darker region to its right, and ii) a binding energy term that models the anatomical relationships between adjacent bones by introducing an elastic force connecting appropriate points of adjacent snakes. The same model and optimization approach developed in [115, 116] is employed in [117] to detect people and segment their body parts (head, torso, legs) in videos. The only difference is the removal of the derivative term and the addition of a new term which allows the model to be applied to color images by considering the gradient of the three components in the color space, as already done in [108]. The method described in [94] segments the brain regions enclosing white matter, gray matter and cerebrospinal fluid, employing a “standard” set of energy terms: continuity and curvature as internal and Kirsch spatial gradient [129] as external. The same authors extend this work in [93], where the internal features are dynamic as they are trained, and therefore may change, from slice to slice, while the trained external features are static and do not change across the whole magnetic resonance dataset. The internal features are the contour length and center. The external features consist of intensity relationships along with the gradient of each contour point and its connected outer neighbor(s). A priori knowledge about the shape of the target object (the mouth) is included in the fitness function in [113, 114]. Lips have a strong intensity level while teeth and the dark interior of the mouth are blurred and rather dark. Thus, a Sobel filter, a binarization, and morphological processing are applied during a preprocessing step. The final aim is to place a snake on the outer lips contour and another on

the inner contour.

In [96] genetic algorithm-based optimization searches for the best location of the contour in two phases: in the first only the internal energy is considered while, in the second, the external energy is also added. Similarly, in [95], a two-stage approach is proposed for solving two main problems: active models' high sensitivity to the choice of the initial contour and the common tendency of numerical gradient-based techniques to get trapped into local minima due to image noise and pseudo-edges. In the first stage, a coarse segmentation of the target object is performed from one image set, slice by slice, using a 2-D method. The latter preliminary segmentation is used as the initial surface. Then, the equation regulating the modification in time of the contour is solved by a finite-difference method, whose result is used to generate the first generation for the parallel genetic algorithm optimizer. In the second stage, the parallel genetic algorithm refines the surface and produces the final result. In order to reconstruct the object surface from the image data using a genetic algorithm, a geometric representation of this discrete surface is used, based on Fourier descriptors.

Another approach that also relies on more than one fitness function is described in [110, 111]. A coevolutionary genetic algorithm adopts a fitness function that consists of several energy terms each having two distinct components: an own fitness, enabling the comparison of different chromosomes within the same population (a candidate solution is represented by a whole population) and a global fitness, measuring the fitness of the chromosome as part of the overall solution. In the former, only the gradient energy is employed. The latter considers three different components: (1) the Euclidean distance between the control points of the current chromosome and the best chromosome in all other populations; (2) a term that minimizes the distance between the end-points of the sub-contour represented by the chromosome under consideration and the end-points of the best chromosome in the neighboring populations; and (3) a term that constrains the contour shape by counting the number of Fourier descriptors of the contour, constructed using the crossing over of the current chromosome with the best chromosome from other populations.

A different evolutionary scheme was implemented in [77]. This method involves the concept of "active contour state" which leads to a multi-stage active contour energy minimization procedure. The contour state is referenced with regard to the  $i^{th}$  state transition using a 5-tuple formulation. For each state a population of different states is randomly created and a genetic algorithm is run. A contour moves into a new state  $i + 1$  when the contour's state  $i + 1$  has a smaller energy than its previous state  $i$ , until no better states are found.

Among the proposals based on swarm intelligence, in [79, 83, 84] particle swarm optimization manages a swarm of particles  $\{O_i\}$ , each of which corresponds to a control point  $p_i$ . The cost of a given particle in the swarm is the local energy of the point. The proposed method iteratively scans each control point, allocates a search window and performs a one-round optimization by minimizing the local energy. When all swarms

become stable the optimization process finishes.

The formulation of the fitness function in those proposals that learn energy term weights through metaheuristics differs significantly from the previous group in many cases. In general, they are based on measuring the segmentation error obtained by the active contour whose parameters were automatically adjusted. To that aim, they use one or more reference images that have been already segmented manually. This is the case of [98], [91], and [101]. Instead of using a segmentation error rate, in [100] the authors use a fitness function based on the distances from the center of gravity (mean of the coordinates) of the manually segmented curve to the  $i^{th}$  point of the automatically segmented curve. Although these evolutionary-based learning paradigms carry out different tasks, a few of them still consider the total energy of the active contour as fitness. This is the case of [102] where there is only one active contour with four control coefficients whose weights are to be learnt by a genetic algorithm. In each generation of the genetic algorithm, the best-fit individual is used for the single-step contour deformation. The fitness function is  $\frac{1}{1+\alpha E_{continuity}+\beta E_{curvature}+\gamma E_{image}+\kappa E_{pressure}}$ .

Finally, in those proposals where the metaheuristic is used during active contour model initialization, the fitness function depends on the specific segmentation task (that determines the global shape, starting position and number of initial curves). In [103], the goal is to obtain the initial candidate curves by optimizing the threshold and  $\sigma$  parameter of Canny edge detector. The fitness function is  $\alpha \frac{\max(L)}{\min(N)}$ , where  $L$  and  $N$  are the length and the total number of pixels of the contour, respectively, and  $\alpha$  is a weight (usually equal to 2/3).

In [81] the authors propose a snake initialization method based on SI, which automatically sets the number of required snakes as well as the appropriate number of control points. Once the swarm reaches the equilibrium, which is tested by continually measuring the activity of the swarm until it drops below a threshold, the agents become ready for shaping up contours. This is achieved by connecting adjacent agents pairwise in a specific order. Each agent chooses two neighbors based on two factors, distance and angle. Concerning distance, closer agents are preferred, while concerning angle, those that form flatter connections. These criteria result in smooth and natural contours. Two thresholds delimit the maximum distance and the cosine of the angle between two connections.

#### 4.1.4. Critical Discussion

Metaheuristic-based approaches applied to active contour models have been shown to be effective when employed as model optimizers, model initializers, and also as the central part of learning paradigms aimed at deriving the best possible energy term parameters for a given ground-truth image set. Apart from the different roles assigned to metaheuristics, the best proof of their success is the wide range of real-world applications described in the literature.

Specific metaheuristic-based active contour model designs have been mainly developed for different med-

ical image segmentation problems. In particular, for segmenting: the foveal avascular zone in ocular fundus images [75]; the optic-nerve head [122]; hand and wrist bones in radiographic images (to study abnormalities in skeletal growth) [116]; the ascending and descending aorta in cardiovascular magnetic resonance images (for measuring regional aortic compliance) [130]; series of histochemically stained cells and laryngoscopic color image sequences [98], breast ultrasound images [102]; tumors in mammograms [119], positron emission tomography images of the liver [103]; teeth in dental computed tomographies [120]; brain magnetic resonance images (to detect white matter, gray matter and cerebrospinal fluid) in [93, 94], and lateral ventricles in [95]. They have also been used for detecting the left ventricle boundary in a sequence of cardiac ultrasound images [86, 91, 96] and angiographic sequences [97, 100]; and the lumen and media-adventitia border in sequential intravascular ultrasound frames [89]. Moreover, the application of these techniques also covers other fields. For example, they have been used for meat analysis to separate connective tissue from meat [108]; in audio-visual speech recognition [113] by lip segmentation; in video surveillance [117] to segment body parts (head, torso, legs); to segment the different components (radicals) of handwritten Chinese characters [123]; in the visual inspection of wood, to automatically detect classical defects such as knots and knotty cylinders [87, 88]; or for tracking moving objects [81, 110].

Other approaches in this category were designed as general-purpose segmentation techniques and tested on different segmentation problems. That is the case of the method presented in [82, 99], which was able to properly segment computed tomographies of the lung and magnetic resonance images of the brain ventricle. Sequences of synthetic images were also included in the test image sets. In [118], the proposed genetic snake was tested on brain tumor segmentation in magnetic resonance, and on synthetic and histological (bacteria and blood) images. Synthetic, computed tomography, and ultrasound images were used as test in [79, 84]. A few papers as [92] just focus on the segmentation of synthetic images drawn *ad hoc* to serve as proofs of concept. Finally, some works established theoretical properties and advantages of the use of genetic algorithms for active contour models optimization [77, 101] but without providing any proofs, either quantitative or qualitative.

The performance of the metaheuristic-based approaches have been evaluated by comparing their results with segmentations manually drawn by experts [89, 91, 96–100, 102, 103, 120, 130], ground-truth data from public repositories [95, 103, 113], software-generated phantoms [92, 99] and/or by comparing them to other active contour model approaches [93, 96, 98, 118, 120, 130], but never with other approaches that use metaheuristics. In almost all those cases the metaheuristic-based approach outperformed its counterparts. However, it is almost impossible to draw an overall conclusion about the relative performance of the different proposals. Almost none of them used a common image repository and they were not compared to other metaheuristic-based active contour models. Some proposals did not even perform any comparison with other segmentation methods [75,



79, 81, 82, 84, 94, 108, 110, 116, 119, 122] providing just a subjective evaluation of their results.

#### 4.2. Active Nets and Volumes

Most approaches based on the active net paradigm hybridize topological active nets or topological active volumes with an evolutionary algorithm. Despite the promising features of topological active nets/volumes, the complexity of the model and the difficult optimization task inherent to the segmentation process have limited their use. In fact, after an early work in 1994 [25], only few approaches based on topological active nets/volumes have been developed in the last ten years. These works have been mainly focused on the use of global optimization techniques. That is why the evolution of topological active nets/volumes models is tightly linked to the evolutionary algorithm-based proposals since the latter solve, at the same time, their main drawbacks and limitations: topological changes, energy terms employed, and the need for an accurate optimization procedure. There are only two exceptions. The first one [27] refers to a novel modelization of the original topological active volumes model [25]. The second one is a recent contribution tackling the topological active nets model's limitations by developing an extension named extended topological active nets [131]. This extended model defines a new strategy for changing its topology, as well as a novel external force and a new local optimization method.

Concerning topological changes, most authors rely on an automatic methodology for topological active net division. The purely evolutionary approach [132, 133] has limitations since the topological active nets/volumes topology cannot be changed to obtain a better adjustment. To overcome this drawback, a memetic algorithm in which an evolutionary and a greedy algorithm are hybridized is employed [134–137]. Following a Lamarckian strategy, all the changes made by the greedy procedure are reverted to the original genotypes. This way, the greedy algorithm allows for topological changes (link cuts and automatic net division) in the net structure to be obtained.

Besides energy terms designed to address specific image segmentation problems, which will be summarized at the end of section 4.2.3, novel energy terms were developed as part of some evolutionary methods. A first example is the gradient distance term  $GD(v(r,s))$ , i.e. the distance from position  $v(r,s)$  to the nearest edge [135]. The gradient distance, whose value diminishes as the node approaches an edge, facilitates the adjustment of the external nodes to the object boundaries. All subsequent publications employed it in both topological active net/volume models. Similar to the previous one, the “in-out distance” term [138] acts as a gradient: for the internal nodes its value decreases as image brightness increases whereas for the external nodes its value decreases with image brightness, ideally reaching its minimum in the background. This term was also extended to the topological active volume model in [139].

Most proposals regard the optimization framework. There is a relevant number of different evolutionary

algorithms hybridized with topological active nets/volumes and they are all used to draw the topological active nets/volumes node locations towards the best possible segmentation. Only a few approaches optimize the topological active net [138, 140] and topological active volume [139–141] energy term weights. Moreover, they do so while, at the same time, searching for the best node locations by means of an evolutionary multi-objective optimization [142] approach. As previously said, all the proposals are based on evolutionary algorithms: genetic algorithms were employed in [133, 135], differential evolution in [136, 140], scatter search in [134], and strength Pareto evolutionary algorithm 2 in [138], among others. However, apart from the initial proposals [132, 133, 143, 144] they all hybridize the global search procedure with the greedy local search proposed in [145] or an extension of the same local search procedure [134]. Finally, a completely different approach has been recently proposed where differential evolution is used to train an artificial neural network [146] that acts as a “segmentation operator” moving each topological active net node in order to reach the final segmentations [147, 148].

Table 3: Active Nets. From left to right: author(s) and reference to the paper, deformable model type, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Name	Dim.	Metaheuristic	Image Modality	Year
Sakaue [149]	Topological Active Net	2-D	Memetic Algorithms	3 stereo images	1996
Ibáñez et al. [132]	Topological Active Net	2-D	Genetic Algorithms	7 synthetic	2006
Ibáñez et al. [135]	Topological Active Net	2-D	Memetic Algorithms	9 synthetic and 4 computed tomographies	2007-2009
Novo et al. [133, 144]	Topological Active Net	2-D	Genetic Algorithms	273 digital retinal fundus images	2008-2009
Novo et al. [138–141]	Topological Active Net and Topological Active Volume	2-D and 3-D	strength Pareto evolutionary algorithm 2 and Differential Evolution	6 synthetic, 2 computed tomographies and retinal (233 + 40) images	2010-2013
Novo et al. [137, 143]	Topological Active Volume	3-D	Memetic Algorithms	6 synthetic and 2 computed tomographies	2009-2011
Novo et al. [136, 150, 151]	Topological Active Net and Topological Active Volume	2-D and 3-D	Differential Evolution	$\simeq$ 20 synthetic and computed tomography images	2011-2012
Novo et al. [147, 148]	Topological Active Net	2-D	Differential Evolution	12 synthetic, 4 computed tomographies	2012-2013
Bova et al. [134]	Extended Topological Active Nets	2-D	Scatter Search	10 synthetic and 10 computed tomographies (knee and lung)	2013

#### 4.2.1. Encoding

For almost all existing approaches each chromosome represents one topological active net/volume. It encodes the cartesian coordinates of every node. An active net chromosome has two genes for each node, one

for its  $x$  coordinate and another for its  $y$  coordinate, both encoded as integer values. Similarly, three real-coded genes encode the  $x$ ,  $y$ , and  $z$  coordinates of each node in topological active volumes. There is only one exception [147, 148], where each individual encodes an artificial neural network: the genotypes encode all the artificial neural network connection weights as real numbers in the range  $[-1, 1]$ .

In addition, since all the memetic approaches consider topological changes in each individual, in those cases a second chromosome encodes the model topology. Therefore, for each node, this additional chromosome encodes the status of the links connecting it to the neighboring nodes (see Figure 5). However, this double chromosome scheme can generate offsprings having crossings in their connections when combining nets with different topologies. The probability of this undesirable event increases with the number of different topologies in the parent nets. The differential evolution-based approaches described in [136, 150, 151] represent the only exception since all individuals share the same topology with the best individual in the population.

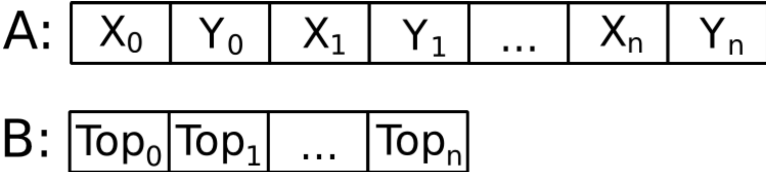


Figure 5: Topological active net encoding.

4.2.2. Operators

From a global optimization perspective, there are several constraints on the topology of the active net models. The same pixel (voxel) cannot correspond to more than one node, link crossings are not allowed, and threads have to be avoided for a proper mesh division (see Figure 6). In many cases, these constraints induced specific designs of the evolutionary operators employed in the corresponding algorithms.

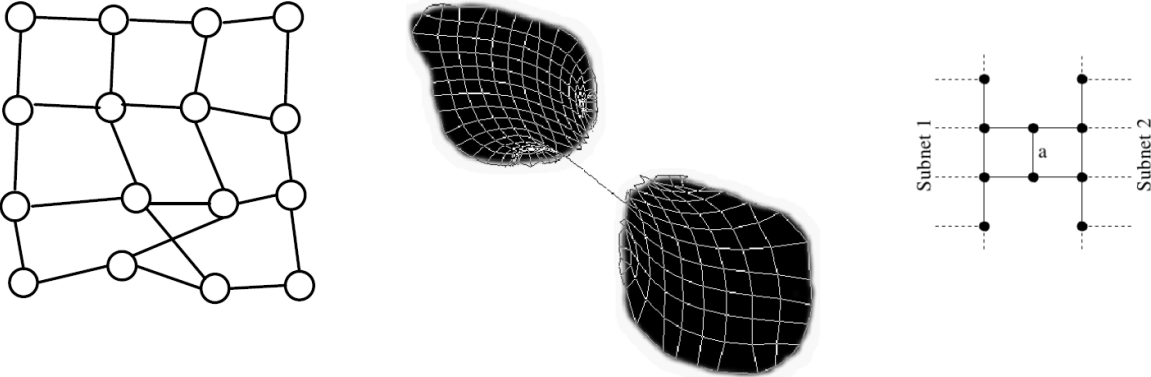


Figure 6: Representation of link crossings and threads in topological active net models.

In particular, most of the existing approaches [132, 133, 135, 137] propose the use of an arithmetic crossover [50] which is only allowed between nets having the same topology, to avoid link crossings in the offsprings. However, this operator is only useful at the very beginning of the search process, since it produces nets having worse fitness than their parents' whenever the search process starts to converge. In addition, it does not incorporate the same information as the parents and may generate unfeasible offsprings when combining nets with different topologies. In the approaches based on differential evolution [138, 140, 147, 148], the classical binomial vector crossover [59] is implemented, where the donor vector is chosen by tournament selection.

More complex crossover operators have been designed in [134] and [149]. In [149], a crosspoint is selected at random ( $C_1$ ) partitioning the active net into four rectangles (see Figure 7). Points  $C_2$  and  $C_3$  are then selected at random within the upper left and lower right rectangles induced by  $C_1$ . Using these points as reference, the crosspoints are combined in fourteen ways resulting in fourteen new network shapes. Among those, the one with the smallest energy is defined as the new individual (child) generated by the crossover.

In [134], two crossover operators are proposed which overcome the limitations of the arithmetic crossover employed in previous approaches (see Figure 8). The genotypic crossover computes a different combination weight  $\theta$  for every pair of homologous nodes of the parent nets. This value is inversely proportional to the local energy of the nodes, in such a way that the location of the corresponding offspring node will be closer to the one corresponding to the parent node with lower local energy. Hence, it works as a heuristic real-coded crossover [50]. The phenotypic crossover computes the binary segmentations of the two parents, then it performs the union of the resulting two binary images and adjusts a mesh to the shape of the object(s) in the union image using the hybridized local search. The resulting net has the same shape of the union of the two parent nets, including a new proper topology calculated by local search.

The mutation operator employed in the majority of the genetic algorithm-based approaches [132, 133, 135, 138] allows a node to mutate to any possible position, without causing any crossing in the topological active net, thanks to an ad hoc design. The basic idea is to compute the area of the four polygons formed by the mutating node and its eight nearest-neighbors, as Figure 9 shows. If the sum of the four subareas is the same before and after the mutation, the mutation is valid and will not cause any crossing. The same idea has been also extended into three dimensions for topological active volume models [137]. The approaches that do not use this operator [134, 149] do not employ any mutation operator at all except for the differential evolution-based approaches [136, 140].

Some approaches also use one or more of the *ad hoc* operators proposed in [135]: the spread operator, the group mutation and the shift operator. They have been designed to maintain the diversity of sizes in the population, to simultaneously modify a group of neighboring nodes in the same direction and by the same

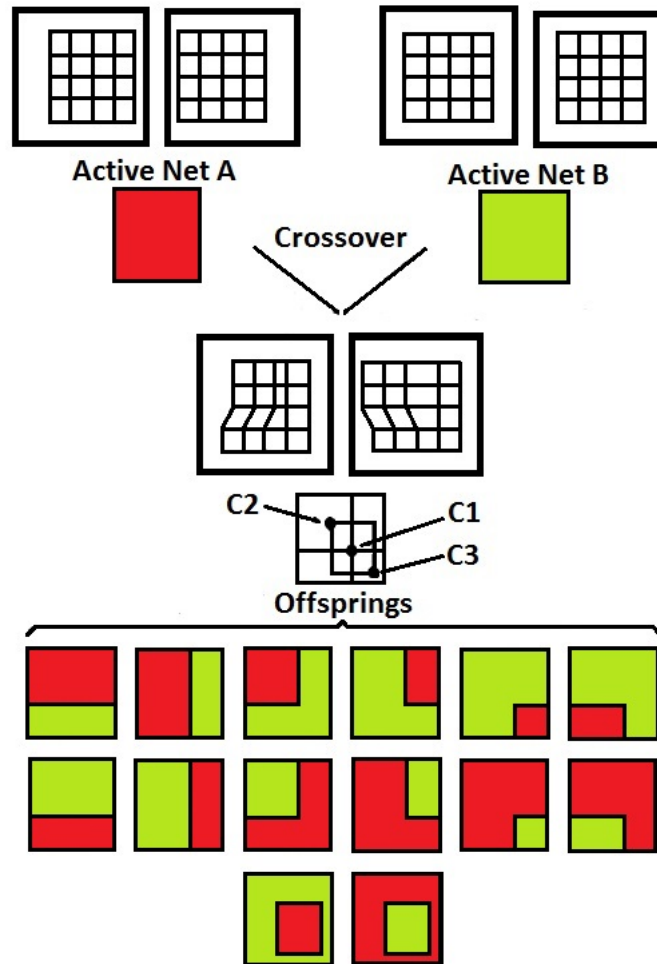


Figure 7: Sakaue's crossover [149].

extent, and to translate the net to another position in the image, respectively.

In contrast to these approaches, which promote diversity through any of those ad hoc operators, the scatter search-based approach by Bova et al. [134] relies on a reference set composed of the best and the most diverse subsets. In particular, the diversity function looks for meshes that properly segment some objects (or parts of them) subject to the a priori requirement that these objects are located far away, on the image plane, from the ones segmented by the nets in the quality subset of the reference set.

Concerning the selection operator, almost all approaches use tournament selection with a size equal to 3% of the population size. We only found two exceptions, the roulette-wheel selection employed in [149] and the two-tier reference set approach followed by the scatter search in [134].

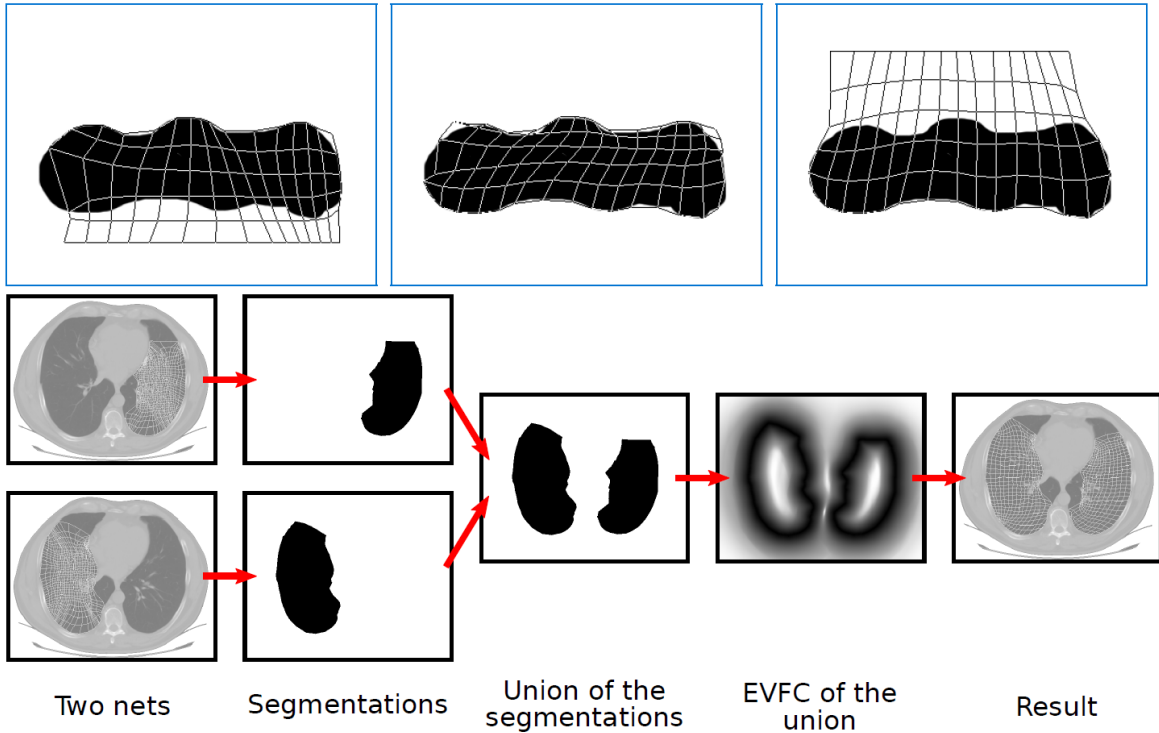


Figure 8: Images in the first row correspond to Bova et al.'s genotypic solution combination operators (parents on both sides and resulting solution in the middle). Images in the second row depicts the phenotypic solution combination process proposed in the same work [134]

#### 4.2.3. Fitness Function

We can say that, in general, all approaches are guided by a fitness function that corresponds to the energy function of the model. Most of them divide the evolutionary process into two stages with two different sets of energy term weights [133, 135, 137]. In the first stage, the energy parameters allow the nodes to be outside the image without a strong penalization so the model can cover the image within the first generations. In the second stage, the parameter values are changed to favor a more homogeneous distribution of the internal nodes, to reduce the size of the net, and to adapt it to the image. Similarly, in [138, 140], an evolutionary multiobjective optimization framework is proposed that, instead of changing the parameter values, uses different objectives in each phase. As an exception, the authors of [147, 148] train a multilayer perceptron model to learn how the topological active net nodes should be iteratively moved in order to reach the desired segmentations. The main purpose of the artificial neural network is to provide, for a given topological active net node, suitable movements that imply an energy minimization of the whole topological active net structure. The fitness associated to each individual (encoded artificial neural network) is the energy of the final segmentation obtained by such an

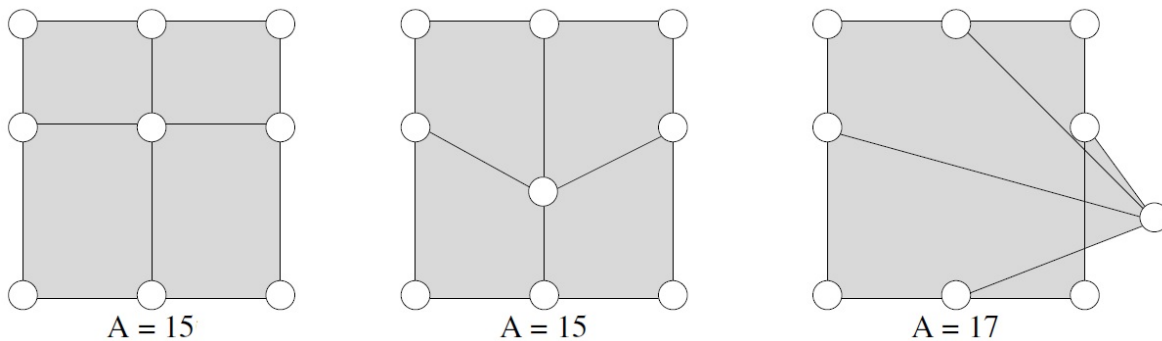


Figure 9: Ibañez et al.’s mutation operator [135]. It uses the area of the polygons formed by the central node (subject to mutation) with its eight nearest neighbors, to check the feasibility of mutation. Left: the four polygons before mutation. Center: a valid mutation; in this case, the sum of the areas remains unchanged. Right: an unfeasible mutation.

artificial neural network.

When tackling image segmentation as an energy minimization task, the correlation between fitness function and segmentation quality plays a critical role. Indeed, if this correlation is loose, even the perfect optimizer leads to suboptimal segmentations. Although the mesh adjustment to the object contour (local search) and the evaluation of a segmentation through its mesh position (global search) are both tackled as minimization problems, they are actually quite different tasks. Hence, it is not surprising that the most suitable fitness functions for the two tasks do not match. A problem shared by the genetic algorithm-greedy memetic approaches is the difficulty to maintain fitness coherence in the population individuals after topological changes in some of them. To deal with that, a node that turns from internal to external after a link cut is only considered as external for fitness computation when it is on the object edges. In contrast, differential evolution- and scatter search-based approaches [134, 136] do not need to employ this mechanism. In the former, all the nets inherit the topology of the best individual. In the latter, two different energy functions are employed. While the same external energy formulation is used for both local and global searches, the internal energy function has been redesigned for the global search in order to solve some specific problems. In particular, the contraction term of the internal energy is not suitable for a global search framework because it penalizes big nets regardless of the size of the target object. Thus, an area-related term has been introduced in the global search fitness function, while the bending term has been removed from it. Authors argue that this term strongly penalizes meshes which split to adapt their topology to the objects, particularly in the presence of many target objects.

Finally, a few approaches use specifically-designed energy terms to address specific image segmentation problems. In [133] a topological active net is employed to detect the optic disc in digital retinal images. To

simplify this task, a new component is introduced that assigns good fitness values to the active nets in the genetic population having a circular structure as the optic disc. Such a term considers the average radius, calculated as the average distance between the centroid of the whole active net and the external nodes. Another term has also been added to favor the correspondence of external nodes with dark pixels and of internal nodes with bright ones, because in retinal images the optic disc has brighter intensities than the surrounding region.

#### 4.2.4. *Critical Discussion*

Few approaches based on active nets [134, 135, 137, 138] have provided solutions to the main drawbacks and criticalities of active nets: sensitivity to topological changes, definition of the external energy, and local deformations. They have already been successfully applied to different problems such as iris location [133] and to the segmentation of different structures in computed tomography images [134–138, 140].

We can draw some objective comparative conclusions about the performance of the different evolutionary topological active net/volume proposals, since the most recent proposals have been compared against previous solutions. However, in most cases, comparisons were made on small image sets and were based on a visual subjective comparison of the results. In [135], the authors evaluate the segmentation results of a greedy, a genetic, and a memetic algorithm over seven synthetic images and three computed tomography images. The evaluation of the results is based on different criteria: from segmentation of fuzzy edges to sensitivity to noise and computation time. The memetic approach outperforms the other two in all categories except in computation time. In [138, 140], the results obtained by the multi-objective approach on synthetic images, computed tomography images of the knee, and retinal images suggest that this approach also outperforms the greedy and evolutionary approaches. Besides its main advantage, the automatic tuning of weights, it produces a net with a more homogeneous distribution of the nodes and is very little sensitive to noise. In [136], the new approach has been tested on a set of 20 images, including 2-D and 3-D synthetic and actual medical images with one or more target objects. The results obtained showed faster convergence and better outcomes than the ones obtained in [135, 137]. Meanwhile, extended topological active nets yield an outstanding performance in comparison with the topological active net model while also outperforming state-of-the-art parametric and geometric deformable models [131]. Finally, the scatter search-based proposal described in [134] has been tested on synthetic and actual computed tomography images of the knee and of the lungs. Its performance has significantly outperformed the one achieved by two extended topological active nets-based optimization approaches (the original local search and a new multi-start local search) and the differential evolution-based memetic algorithm [136].

Although the segmentation results obtained by all the evolutionary approaches improve on the greedy approach, their applicability is still limited when dealing with real-world images and complex synthetic ones. In



particular, those algorithms fail to design evolutionary operators able to effectively combine nets and consequently require very large populations. In some approaches, the population size is around 1000 individuals (even 2000 in the case of topological active volume [140]), each encoding a complete mesh and the corresponding topology. Consequently, the search needs to be run for thousands of generations. In addition, they lack a proper energy definition for a global optimization scenario. Only the recent work based on scatter search [134] has overcome these two important limitations.

### 4.3. Statistical Shape Models

One of the most populated categories within deformable model research includes models that use some sort of statistical information about the objects to segment instead of using only generic constraints, as active contour models do. Such deformable models can be created by running some mathematical transform (like principal component analysis [152], approximated principal geodesic analysis [153], or hierarchical regional principal component analysis [154]) on a number of training shapes, or by including more information specific to the deformation shape limits. In all these cases, we will refer here to active shape models, active appearance models, deformable templates and similar approaches by the common term of statistical shape models, even if the contributions where they are described do not refer explicitly to such a category.

The approaches described in this section aim to optimize the positions of the control points of the deformable model, i.e., metaheuristics are employed to “guide” the movement and deformation of the deformable model [153, 155–161], to optimize/tune different parameters of the segmentation method [162, 163], or, most frequently, to optimize the weights of the main modes of variation found in the training set and the parameters of a (usually affine) transformation [152, 154, 164–169].

The potential of applying metaheuristics to this kind of deformable models was discovered rather early. Five papers were already presented in the 90’s where deformable models were used in combination with metaheuristics. Since the seminal papers by Hill et al. [170, 171], metaheuristics have been frequently used, alone or in hybrid approaches, as optimizers to overcome the problems of classic methods, mainly related with deformable model initialization, parameter selection, and existence of local optima.

All works remark the interest for applying metaheuristics, in consideration of the multimodality of the problem, the lack of knowledge about the mathematical properties of the objective function to optimize (continuity, differentiability, convexity,...), and the nature of metaheuristics, which avoids the need for computing derivatives, which is often a time-consuming task, can involve approximation errors or be impossible at all.

#### 4.3.1. Encoding

In general, the parameters to be optimized by the metaheuristic are either the coordinates of the control points of the parametric deformable model [157] or the coefficients of a parameterized curve (like a B-

Table 4: Statistical Shape Models (Active Shape Models, Active Appearance Models, Deformable Templates). This table continues as Table 5. From left to right: author(s) and reference to the paper, deformable model type, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Name	Dims.	Metaheuristic	Image Modality	Year
Hill et al. [171]  [170]	Flexible Templates (like Active Shape Model)	2-D	Genetic Algorithm vs Simulated Annealing	ultrasound heart's left ventricle 20 and 25 images	1992
Jolly et al. [155]	Deformable Template	2-D	Simulated Annealing (Metropolis algorithm)	393 image sequences moving vehicles	1996
Rueckert & Burger [130],[168],[169]	Geometrically Deformable Template	4D	Simulated Annealing + Iterated Conditional Modes	1 magnetic resonance image sequence	1996, 1997
Mignotte et al.[166]	Deformable Template / Active Shape Model	2-D	Genetic Algorithm + steepest ascent	echographic sequence (50 frames) & synthetic images	2001
Pitiot et al.[167]	Statistical Shape Model	2-D	Evolutionary Simulated Annealing	magnetic resonance images 94 images and 3 brain structures	2002
Lalonde & Gagnon[172]	Geometrically Deformable Template	2-D	Variable Neighborhood Search using Simulated Annealing as local search	optic disc in 2 ophthalmic images	2002
Betrouni et al.[164]	Active Appearance Model	2-D	Simulated Annealing	30 abdominal ultrasound images	2004
Karungaru et al. [173]	Deformable Template	2-D	real-coded Genetic Algorithm	54 faces	2004
Vavilin & Jo[174]	Deformable Template	2-D	Genetic Algorithm	95 images with 119 signs	2006
Heimann et al.[152]	Statistical Shape Model + Deformable Surface	3-D	Evolutionary Strategy	liver in 54 computed tomography volumes	2007
Sattar et al. [175],[176],[177]	Active Appearance Model	2-D /3-D	Genetic Algorithm vs Nelder-mead simplex vs Non Sorting Genetic Algorithm II Gradient-based Genetic Algorithm	face images 32 subjects 1405 images real & synthetic images	2008
Mata et al.[163]	Deformable Template	2-D	Genetic Algorithm	real-time video sequences	2008
Yin & Collins[178]	Deformable Template/ Active Shape Model	2-D	Simulated Annealing	aerial video of motorcycle bladder magnetic resonance imaging human walking sequence	2009
Chen et al.[179]	Active Shape Model	2-D	Genetic Algorithm	20 lateral knee X-ray images	2009

Table 5: Statistical Shape Models (Active Shape Models, Active Appearance Models, Deformable Templates). This table is the second part of Table 4. From left to right: author(s) and reference to the paper, deformable model type, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Name	Dims.	Metaheuristic	Image Modality	Year
Szilagyı et al.[180]	Active Appearance Motion Model	4D	multipopulation Genetic Algorithm + Nelder-Mead simplex	heart ultrasound Number not specified	2009
Takahashi et al.[181]	Active Shape Model	2-D	Genetic Algorithm	5 computed tomography images cardiac region	2010
Colutto et al.[153]	3-D medial axis shape model	3-D	CMA-ES (vs Particle Swarm Optimization and Differential Evolution)	3-D magnetic resonance imaging scan & synthetic data	2010
Ben Fredj et al.[162]	Deformable Template	2-D /3-D	Genetic Algorithm	synthetic heart 3-D magnetic resonance imaging	2011
McIntosh et al.[165],[154]	Medial-based shape	2-D	Genetic Algorithm	magnetic resonance imaging 50 images of corpus callosum	2010, 2012
Mussi et al. [182],[158],[159] Ugolotti et al.[183]	Deformable Template	2-D	Particle Swarm Optimization Differential Evolution vs Particle Swarm Optimization	road signs human body pose road signs	2010, 2012 2012
Liu te al.[184]	Active Appearance Model	2-D	Genetic Algorithm	200 tongue images	2012
Mesejo et al. [157], [156] Ugolotti et al. [161], [160]	Active Shape Model	2-D	Differential Evolution vs Particle Swarm Optimization vs Levenberg-Marquardt vs Simulated Annealing vs Scatter Search vs Genetic Algorithm	histological brain (hippocampus) 320 real images + 20 synthetic images + 4 human video-sequences	2011, 2012 2013

Spline) [167]. Such coordinates usually refer to a Cartesian reference system but polar coordinates are also used [160].

In [170, 171], the model parameters for the segmentation of the heart's left ventricle are encoded as ten unsigned binary integers. These parameters control the shape and transformations of the model and include six shape parameters and four transformation parameters (accounting for translations along the  $x$  and  $y$  axes, rotation and scale). In [179, 181], a similar approach is carried out in terms of deformable model representation and encoding.

In [167], an individual in the population consists of pose and shape parameters, including scale, rotation, translation and the weights for the first  $m$  eigenvectors (they take  $m = \text{number\_of\_training\_instances}/10$ ). In this case, the evolutionary algorithm uses prior statistical information about the shape of the target structure to control the behavior of a number of deformable templates, each template being modeled in the form of a B-spline. The population is made up of  $f$  families, and each family  $F^i$  consists of a parent  $P^i$  and  $c_i$  children  $C^{i,j}$ .

In [154, 165], unlike in boundary-based techniques, a medial-axis-based 2-D shape representation is used. This kind of representation describes the object's shapes in terms of an axis (the medial-axis) with respect to which the object is approximately symmetric, along with thickness values assigned to each point along the axis, describing the object boundary shape. A real-valued encoding is used and each shape is represented as a chromosome whose genes encode affine and statistical shape deformation parameters. Affine transformation parameters encode global rotation, scale, and position of the shape. Statistical shape deformation parameters represent the weights of the principal components, obtained using hierarchical regional principal component analysis for a particular deformation, location, and scale.

Besides these, a sort of medial-based representations can be found in [156–161, 182, 183] where, even if according to very different schemes, the common factor is the use of simple skeleton representations of the object to localize, segment or track. Of course, this kind of minimal representation has many advantages, like a low computation cost allowing for real-time execution, the simplification of the operations on the coordinates and of their constraints, and the opportunity to perform a fast initial coarse localization and, later, a finer segmentation. In particular, in [156, 157, 160], the encoding includes lengths and angles from one point in the model to another, i.e., the relative position of the control points (joints), represented in polar coordinates. In [158, 159, 161], an articulated 3-D model of a human body is matched against the frames of a set of videos of a human performing some action, taken from different perspectives, to estimate the subject's posture in space and time. The body model consists of two layers, the skeleton and the skin. The skeleton is defined as a set of cascaded homogeneous transformation matrices which encode the information about the position and orienta-

tion of every joint with respect to its parent joint, in a kinematic tree hierarchy. The skin layer is connected to the skeleton through the joints' local coordinate systems and each joint controls a corresponding skin region. In all these cases, a real-valued encoding was used by the metaheuristic and the search space was the space of all plausible skeleton configurations where every individual is composed of  $3 + M \cdot 3$  genes, corresponding to the position of the root joint with respect to the reference (world) coordinate system and the rotational degrees of freedom of each of the  $M$  joints around every axis in the three dimensions.

A classical representation is used in [152, 164] where the deformable model is represented by a point distribution model, i.e., a dense collection of landmark points on the object's surface. Each training shape is described by a single vector of concatenated landmark coordinates. In [152], an individual, stored as a real-valued vector, represents one of the possible shape configurations, consisting of a similarity transform and several shape parameters. The pose parameters (translation, rotation, scale) are estimated from the mean values of the corresponding training samples (using relative coordinates for location) and also randomly generated according to a Gaussian distribution. In [164], the automatic segmentation of prostate boundaries from abdominal ultrasound images is performed, and every real-coded chromosome represents parameters describing the variations of the prostate boundary with respect to an average contour.

In [155], the authors define a polygonal template to characterize a general model of a vehicle and derive a prior probability density function to constrain the template to be deformed to lie within a set of allowed shapes. Every solution encodes a deformable template, characterized by a finite set of parameters representing the vertices' locations.

Another Bayesian formulation is presented in [168, 169] and solved using simulated annealing. The authors present an energy-minimization framework based on geometrically deformable templates. To solve the segmentation problem it is necessary to estimate the optimal values for the following parameters: translation, scaling, rotation, and  $N$  parameters for the non-affine deformation. The affine values are calculated by an exhaustive search method (iterated conditional modes). The non-affine ones are estimated using simulated annealing.

A cubic B-spline shape representation based on a binary encoding is used in [166]. In this case, the set of feasible deformations includes translation, scaling, rotation, and stretching of the template. Thus, the parameter vector to be optimized contains five parameters corresponding to the affine transformations, four pairs of global non-affine deformation parameters, and  $\tau - 2$  pairs of translation vectors,  $\tau$  being the number of control points.

In [163], genetic algorithms are used in three different stages: a preliminary region of interest identification stage and two learning stages. In the first, each individual's genome is made up of five variables: the deformed model's Cartesian coordinates  $(x, y)$  in the image, its horizontal and vertical size in pixels, and a measure of its vertical perspective distortion. Then, during a first training step, a second genetic algorithm, that searches the

training image space, is used for learning the best threshold values for the color segmentation of a landmark, i.e., the target object bounded by a rectangular box. Thus, each individual in this genetic algorithm encodes upper and lower thresholds for the HSL color components (six values). Finally, during the second training step, a third genetic algorithm is used to find the values  $d_0, \dots, d_3$  that determine the best position for a pattern-window that is placed over the model diagonals to exploit the frequent symmetries which can be observed in the objects used as landmarks.

A real-coded genetic algorithm is used to select the best deformable model parameters in [162]. The main idea is to find relevant ranges in a large parameter space, including external force factor, parameters controlling the template stiffness, coefficients describing the model compressibility, and number of iterations of the deformation process.

In [153], a parametric 3-D medial-axis representation (3D-MRep) [185, 186] is used as shape model. The parameters defining the 3D-MRep shape are encoded and optimized using CMA-ES. The parameter domain of 3D-MReps constitutes a finite-dimensional Riemannian manifold and each element of this shape space defines a surface. 3D-MRep uses a discretization of an object's continuous medial axis instead of boundary representations. Every instance of an MRep in three dimensions is represented by a regular mesh described by the coordinates of its vertices (called atoms). A detailed description of the information encoded for every atom can be found in [153].

In [173], a genetic algorithm is employed to reinforce the segmentation results (lip and eye regions) by aiding the matching of the general shapes with the segmented areas. A 26-gene chromosome is used. The first 10 genes represent the height and width of the shapes. The remaining 16 genes represent selected points in the shape (at roughly equal distance from each other).

#### 4.3.2. Operators

In general terms, the metaheuristic operators are generic and they do not present a great level of sophistication because the statistical shape models are usually already constrained within a specific region of the search space as the result of a training stage.

In [167], a hybrid evolutionary algorithm inspired by guided evolutionary simulated annealing [187] is used. The population is made up of families and each family consists of one parent and its children. The pose parameters of each of the parents are selected at random (their shape being the mean shape). The children in each family are generated by randomly modifying the pose parameters of the parents and choosing valid instances of the shape model. The authors introduce restrictions to ensure that the deformed template is a valid instance of the shape model. The main components used in the algorithm are: i) a local search, where the parent and the children are evolved according to a local deformation template matching; ii) a local competition, where the

child with best fitness is selected and may replace its parent following a simulated annealing scheme; iii) a shape constraint imposed on the projection of the new parent onto the shape space; and iv) a second level competition among families. In this approach, to favor the children whose shapes are closer to the first eigenmodes, the temperature of the Boltzmann distribution that controls the acceptance criterion is varied. At a local level, the children of the same family compete with one another to generate the parent for the next generation. In a second level, there is a competition between the families, since the number of children allocated to each family depends on the combined fitness of all their children, and is biased toward the first eigenmodes to favor the most likely shapes. The number of children actually reflects the relevance of a given area of the search space. The better the candidate solutions in a given area, the more attention it attracts. Therefore, the entire algorithm can be viewed as a parallel simulated annealing with competition.

In [170, 171] a standard genetic algorithm is used to optimize shape models (using one-point crossover and bit string mutation). The remainder stochastic independent sampling algorithm [188] is used as selection operator. Niching is also applied to avoid premature convergence to suboptimal solutions. The fitness of an individual is weighted by the number of neighbors (the more the neighbors of an individual, the worse its fitness), the size of the population is allowed to increase, and a restricted mating strategy is implemented in order to promote speciation, i.e., neighbors are preferred to distant individuals for crossover. In [180], the optimization problem is also solved by means of a multi-population genetic algorithm approach, which creates subpopulations within the niches defined by multiple potential optima. Subpopulations only interact by means of a migration process where the best individuals from one subpopulation are copied into another subpopulation, replacing the worst individuals in the destination search niche. Intensification is achieved by allocating a separate portion of the search space to each subpopulation. After convergence of the global search, a Nelder-Mead simplex algorithm [189] is further applied starting from the best solution. Each subpopulation uses scaled fitness-proportionate selection, uniform crossover, and random mutation.

In [154, 165], mutation is performed by altering some weights of the hierarchical regional principal component analysis while crossover consists of swapping a set of weights between two individuals (see Figure 10). Roulette-wheel selection is used along with a coarse-to-fine approach to steer the evolution, facilitating faster initial convergence. Initially mutation is applied only to the affine transformation parameters (2 translation values, 1 model orientation angle, and 2 scaling values). After that, the authors allow mutations to start including shape deformations. Dynamic mutation of a single gene amounts to altering the corresponding weight by sampling it from a uniform random distribution under the constraint that the total weight lies within  $\pm 3$  standard deviations of the corresponding mode of variation.

In [156, 157, 160], the restrictions, in terms of deformation, are managed by the polar representation and

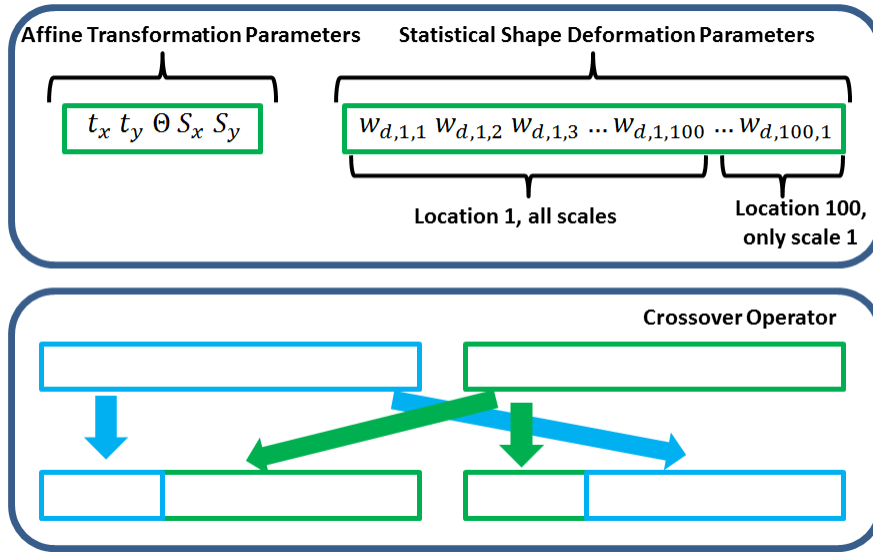


Figure 10: In [154], like in many other statistical shape model approaches, segmenting an anatomical structure consists of finding the optimal set of shape parameters. The authors represent each shape as a chromosome with genes encoding affine and statistical shape deformation parameters.

the limits for the deformation are calculated from a training set. Thus, the conventional operators for genetic algorithms (scatter search, differential evolution, simulated annealing, and particle swarm optimization) are used. In [152], an evolutionary strategy is run to find a rough initialization in a strongly down-sampled version of the image. There is no crossover operator and solutions are mutated by adding a random vector from a multivariate zero-mean Gaussian distribution (Gaussian mutation). In [176], a genetic algorithm searches faces globally whereas gradient descent helps the genetic algorithm to search a face locally. In other words, the exploitation properties of gradient descent and the exploration properties of the genetic algorithm are combined into an effective memetic optimizer. For this reason, the authors propose a new gradient operator which operates in conjunction with the mutation operator already available in the genetic algorithm.

In [166], 100 individuals (each representing an ellipse: circle + affine transform) are initially randomly generated in the image when segmenting ecographic sequences. A genetic algorithm with one-point crossover operator, elitism and a local optimization procedure is used. In each generation, 5% of the best individuals are selected for hybridization with the local optimization technique (gradient ascent).

In [172] two modifications to the geometrically deformable template model are proposed to perform the segmentation of the optic disc in ophthalmic images. The proposed search method is computationally costly and the invariance to affine transformations may cause the search algorithm to retain invalid solutions (e.g. ellipses when searching for circles). The authors address these concerns using a metaheuristic called Variable Neigh-



neighborhood Search, that treats simulated annealing as a local search tool, and also by redefining the shape energy so that affine transformations are taken into account to improve search quality. The main operators used in this metaheuristic are the shaking step, that chooses a random configuration of points within the neighborhood, the local search, and the moving step in which, if the local search finds a configuration with lower energy than the current best solution, the new configuration is set to be the starting point for another step.

#### 4.3.3. *Fitness Function*

The most common approach used in this family of methods aims to maximize the overlap of the deformable model with the object of interest. It takes into account intensities, boundaries, or textures, and controls the feasibility of the new configuration using the main modes of variation computed from a training set. All these features are introduced as terms in the deformable model energy formulation that is directly taken as the fitness function to be minimized. Prior knowledge about every particular case can be very easily included in the fitness function, as happens, for example, in [157]. In that work, since the target anatomical district (the hippocampus) is slightly darker than the immediately surrounding area, the fitness function includes a term (to be maximized) which is proportional to the difference in intensity between the inner model points (ideally corresponding to the hippocampus) and the outer model points (ideally corresponding to the external, clearer regions surrounding the hippocampus). Along with the previously introduced external energy, the model (as well as the fitness function) includes an internal energy term, based on the shape deformation limits found in a training set, that controls the maximum amount of distortion the deformable model can present.

In [158, 159, 161], the fitness function compares the silhouettes extracted from the images to be processed to the silhouettes generated by the model in its candidate pose. The lower the fitness value, the closer the candidate pose to the model position. In [167], the fitness function also measures the matching between the deformed template and a modified edge image, and the elastic deformation energy required in the warping process. The compound functional to be minimized includes stretching and bending energies, along with an attraction factor which favors the attraction to the image edges. In [175–177], fitness is calculated pixelwise as the difference between the deformed curve (obtained by the appearance parameters of each chromosome) and the test image.

In [155], simulated annealing optimizes the vertices' positions (the authors do not specify if those positions are encoded as real or integer numbers). The energy function is minimized by constructing a sequence of template deformations starting from a prototype template. In each iteration, the algorithm determines a new value for the deformation parameters, based on their current one. The authors propose a likelihood probability density function which combines motion information and edge directionality to ensure that the deformable template is contained within the moving areas in the image and that its boundary coincides with strong edges

with the same orientation in the image. The probability density function is modeled as a Gibbs distribution whose exponent comprises two terms. One term is a function which derives from the motion of the vehicle of interest and is maximized when the deformed template encompasses only pixels that are moving. The other term is a directional edge-based function that is maximized when the contours of the deformed template coincide with underlying image edges that have a strong gradient magnitude and whose gradient orientation is perpendicular to the contour. By using the gradient magnitude and direction on the template boundary, the authors implicitly correlate both sides (object and background) of the boundary. Since the energy function features many local minima, simulated annealing is applied according to a geometric cooling scheme.

In [166], the fitness function is composed of a likelihood energy term that is minimized when the deformed template delimits exactly two homogeneous regions with gray level distributions corresponding to blood and muscle. A prior energy term that penalizes the deviation of the deformed template from the original prototype is also included.

A model of the heart is composed of three different objects in [168, 169]. The fitness function to be optimized by the metaheuristic contains an internal and an external term. The former measures the amount of non-affine transformation of the template with respect to the equilibrium shape. The latter is a form of potential that attracts the template towards specific image features, such as edges.

In [153], the authors investigate two approaches for detecting objects in voxel images based on the 3D-MRep shape models [185]. One involves a simplified Mumford-Shah Functional used to segment synthetic 3-D images. The other is an edge-based segmentation for the cerebellum. In the former, the authors search for sets of voxels representing volumes characterized by a significant difference between the mean image intensity inside the volume and the mean intensity of the background. If the objects of interest have nearly the same intensity as the surrounding volume, region-based segmentation cannot be applied. To overcome this problem an edge-based segmentation method is introduced in the latter. In that case, a functional with two terms is used: the first term forces a Jordan submanifold to be at locations where the gradient of voxel data is high, and the second one penalizes the surface area of such a submanifold. The authors use CMA-ES to minimize the functionals.

#### *4.3.4. Critical Discussion*

In general terms, there is a relevant number of publications dealing with statistical shape models. With regard to the strengths of these approaches, metaheuristics increase the flexibility of the deformable model design. They allow one to ignore the nature of the objective function (differentiability, convexity) and to use the method as a black-box optimizer where different combinations of parameters, usually principal component weights and transformation parameters, are tested in an inherently parallel manner, typical of many metaheuristics. Metaheuristics are also used to guide the deformable model deformation and movement, as well as to optimize its

parameters, although the latter option is not too common in statistical shape models. Regarding the weaknesses found in the different approaches, we could mention the following:

- In general terms, the contributions published so far do not provide a clear justification for the use of metaheuristics and it seems that, in some cases, a simpler local optimizer or classical optimization method could have worked properly as well.
- With regard to the former point, comparisons with traditional methods are seldom reported, even if they would be really useful and illuminating. Those comparisons could demonstrate the effectiveness of introducing metaheuristics instead of using more traditional methods.
- Finally, in the vast majority of the papers, there is no detailed description and presentation of the parameters and operators used, there is no proper statistical comparison of results, while no information about the parameter tuning procedure followed is provided.

In terms of applications, a broad variety of problems has been tackled, ranging from face/body recognition [158, 159, 161, 173, 175–177] and car/road-sign localization [155, 174, 178, 182, 183] to medical image segmentation [152–154, 156, 157, 160–162, 164–172, 179–181, 184], or mobile robot navigation [163]. In particular, in this section we have discussed Bayesian approaches [155, 166, 168, 169], the online definition of a deformable model for mobile robot navigation [163], and approaches focused on the segmentation of objects using sophisticated metaheuristics (a generalization of the CMA-ES on vector spaces to Riemannian manifolds [153], as well as an evolutionary version of simulated annealing [167]), or the appropriate use of evolutionary algorithms in the solution of very difficult biomedical problems [154, 157].

#### *4.4. Level Set Method*

The first remark worth making about this sort of techniques is related to the shortage of proposals where geometric/implicit approaches are used in conjunction with metaheuristics. We could find only 16 papers of this kind, four of which apply practically the same basic design [190–192]. In other cases, the use of metaheuristics has not been adequately justified or explained [193–195]. Most often, classic optimization methods, like gradient descent, are used. However, in some cases, like geodesic active contours, such techniques have been shown to be ineffective and likely to produce suboptimal solutions. Moreover, many deformable models are based on partial differential equations which can be solved by traditional numerical methods. However, metaheuristics have demonstrated to be very useful for learning the parameters of the model [196–200], to refine the results obtained by the geometric approach [201], to initialize the contour and/or extract the prior information which is to be used by the level set method [195, 198, 202] or to directly guide the optimization process avoiding local

minima [190–193, 203]. An important advantage of using metaheuristics is that they can optimize the level set function without the need to compute derivatives, thereby permitting a straightforward introduction of new curve-evolution terms [198]. Moreover, in [202], metaheuristics are used along with a parametric deformable model to initialize the level set contour.

In several papers where authors have used eigenshapes (the main modes of deformation computed from a training set of shapes), despite their differences (kind of textural measures used and calculation of the affine transformation), the fitness function and the general pipeline is essentially similar. All proposals employ texture and shape information to evolve the contour using a training and a test phase. The fitness of a given shape is determined by the matching degree between the texture of its enclosed region and the mean texture of the target object calculated in the training phase.

Table 6: Level Set Method. From left to right: author(s) and reference to the paper, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Dims.	Metaheuristic	Image type	Year
Wang et al. [194] [204]	2-D	Evolutionary Strategy	1 cell image, 1 magnetic resonance image 1 computed tomography, 1 ultrasound (9 imgs)	2002 2004
Xiao et al. [195]	2-D	Genetic Algorithm	1 Histological 1 Mammogram	2005
Ghosh et al. [190, 191] Mesejo et al. [192]	2-D-3-D 2-D	Genetic Algorithm Differential Evolution vs Particle Swarm Optimization vs real-coded Genetic Algorithm	10 prostate computed tomographies [190], 100 computed tomographies and 100 magnetic resonance imgs. [191], 10 histological imgs. [192]	2006 2013
Kan et al. [193]	2-D	Particle Swarm Optimization	4 Tree images	2007
Iakovidis et al. [197]	2-D	Genetic Algorithm	Ultrasound images	2007
Law et al. [203]	2-D	Basin Hopping vs Gradient Descent vs Simulated Annealing	1 Natural scene 1 Brain magnetic resonance imaging 1 Zebrafish Intestine Image 1 Breast cancer cell image	2008
Heydarian et al. [196]	2-D	Dynamic Genetic Algorithm	4 computed tomography imgs. (kidney and lung) 2 magnetic resonance images (kidney)	2009
Oliveira et al. [199] [200]	2-D	Genetic Algorithm	20 liver computed tomography	2010 2011
Feltell & Li[201]	3-D	swarm agents	1 brain simulated magnetic resonance (181 imgs)	2010
Mesejo et al. [198]	2-D	Scatter Search and Genetic Algorithm	26 Microscopy, 17 magnetic resonance images and 10 computed tomographies	2013
Mesejo & Cagnoni[202]	2-D	Differential Evolution	26 Histological Images	2013

#### 4.4.1. Encoding

Genetic algorithms have been used in the majority of methods. In some cases, they rely on a binary encoding [196, 197] even when the features of the problem may suggest using a real-number representation.

In the works of Ghosh et al. [190, 191] and Mesejo et al. [192] the parameters that are optimized are the coefficients/weights of a linear combination of the eigenshapes. Some approaches [190, 191] also consider the pose of the object to segment (position, size, and orientation) and encode it as the parameters of an affine transformation of that object. In other cases, for example, the parameters represent the location (in polar coordinates) of the points of a parametric model that is used later as initial boundary for the level set [202], the weights for the different terms in the level set equation [197, 198], or the parameters which encode prior knowledge [195, 198].

In [201], the segmentation problem is treated as a fuzzy voxel classification along the level set interface but the presence of anisotropic voxel intensities within the input image causes an unacceptable number of incorrectly classified voxels. To solve this problem, after the user identifies the area of interest, intelligent semi-autonomous agents move across the zero-level surface and modify it. The level set representation allows the authors to locate the surface in space and compute the normal to the surface in any point very easily. Therefore, the encoding used by the swarms of intelligent agents is the same as the representation used by the level set (the voxel coordinates). The agents inhabit the surface, modifying the sparse field, and update the values of all zero-level points within a given range  $r$  of the agent's position  $x$  by a value  $a = A(x)$ , weighted by the distance from the agent, according to a normalized Gaussian distribution.

#### 4.4.2. Operators

In general, the operators used in level set-based methods are the traditional ones, usually not specifically customized to the problem at hand. The model constraints are usually managed within the level set formulation. The curves are split and merged naturally adjusting the object topology. Furthermore, in some cases, the existence of a training set already constrains the possibilities of generating new shapes. In some cases, the authors use operators which are not particularly well-suited to the nature of the chromosomes, like in [190, 191] where, despite having a real-valued genome, single-point crossover is used.

In [201], agents inhabit the zero-level surface, sensing and modifying it as necessary using straightforward interpolation routines. They are allowed to modify the surface at their location whilst maintaining the structure of the sparse field. Basically, every agent moves independently using two rules/operators: a) navigation over the surface, and b) modification of the sparse field. For the former, a movement potential function  $P$  needs to be defined, while for the latter it is necessary to define a surface update function  $A$ .

In [203], the authors have designed a method based on two operations: hopping and local optimization. In turn, the hopping phase consists of a stochastic split and a deterministic merge, in which both splitting and

merging of a large area of the image may occur in each step of this region-based segmentation algorithm, reducing the number of hops needed to search for a global minimum. To split the selected segment into two sub-segments, a thresholding method [205] is used. The split-and-merge step is the crucial contribution in this algorithm. After merging, the local optimization operator applies the gradient descent method. If the resulting energy is decreased (compared to the energy before the splitting step), then the new configuration is accepted and another round of hopping is performed. Otherwise, the system reverts back to the configuration previous to splitting, remove the last chosen segment from the list of candidates for splitting, and re-normalize the probability of choosing each segment.

In [204], only mutation is used, obtained as an additive Gaussian perturbation because the recombination of contours requires a much higher computational complexity. As selection scheme, a modified  $(\mu, \lambda)$  is applied. Since the acceptance of a temporary deterioration might make also  $(\mu, \lambda)$  selection drift away from the contour energy minimum, the authors select the configuration with lowest energy from  $\mu$  survivors in  $(\mu, \lambda)$  selection, and compare it to the lowest energy configuration among those previously selected. The lowest-energy configuration found in all the selections represents the output contour and the solution to the problem at the end of the algorithm.

Finally, in [206], a dynamic genetic algorithm is used that adaptively changes mutation and crossover probabilities, as well as the number of crossover points.

#### 4.4.3. *Fitness Function*

Two of the approaches that take most advantage of metaheuristics can be found in [198] and [203]. The former describes a hybrid geometric deformable model, combining region- and edge-based information with the prior shape knowledge introduced using deformable registration. Such an approach implies the learning of the level set parameters by means of a genetic algorithm and the use of scatter search to derive the shape prior. In the latter, the computation of a global minimum of the Chan-Vese model is performed combining gradient-descent with a stochastic optimization phase which allows the search to hop from a local minimum (basin) to another, while the computation cost is alleviated using a multiresolution approach. In a single hop, a large area in the image is split and merged. Energy increases after the split-and-merge but decreases significantly after the gradient descent iterations.

In [204], an extension of the work presented in [194], the authors take the viewpoint that region information can be introduced as extra constraints within the contour energy-minimization framework. The contour energy minimization problem is thus formulated as the search for a minimum-energy contour with its interior satisfying a region-based constraint. Such a constraint can be any function characterizing the contour inner structure. In this case, the authors adopt a criterion based on region homogeneity. The introduction of the constraint is

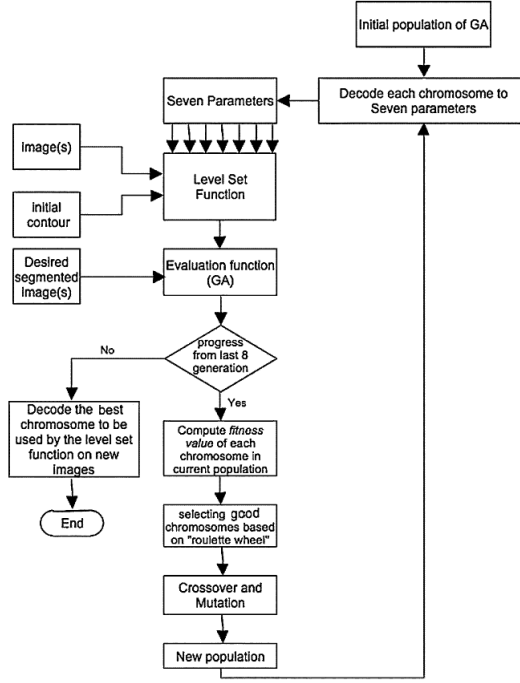


Figure 11: Main workflow diagram for [196].

aimed at limiting the search space of contours, focusing on those with desirable interior properties, while an evolutionary strategy is used to solve the energy minimization problem. Possibly, with this constrained contour formulation, a traditional optimization approach could have solved the problem as well.

The main aim in [196] is to select the optimal values for the seven parameters introduced in [206] and eliminate the need for level set re-initialization for different kinds of image modalities (computed tomography and magnetic resonance imaging in particular) and organs (see Figure 11). To do so, the authors designed an evaluation function comprising four measures to calculate geometric differences between the object boundaries as determined by the level set method and the desired object boundaries. A similar approach is used in [198] where a genetic algorithm is in charge of tuning the weights and the parameters of each term based on training data. In general, the quality of a solution is defined as the average quality of the segmentations obtained in the training phase using the parameter values it encodes.

In [204], the problem is to find a closed contour  $C(s, t)$  enclosing a region  $\Omega_C$  such that

$$E(C(s, t)) = \frac{1}{\oint_C ds} \left\{ \oint_C \frac{1}{1 + |\nabla G * I(x(s), y(s))|^p} ds \right\} \quad (4)$$

is minimized subject to the constraint characterizing the region

$$D(x, y) = \frac{1}{1 + |\nabla G * I|^2} e^{-\frac{|I(x, y) - I_0|}{\sigma}} \geq T_V \quad (5)$$

for all  $(x, y) \in \Omega_C$ , where  $p = 1$  or  $2$ ,  $T_V$  is the similarity threshold, and  $|\nabla G * I|$  is the absolute value of the image intensity gradient ( $I(x, y)$ ) smoothed by a Gaussian filter  $N(0, \sigma_0^2)$ . The solution to the constrained optimization problem is to use an evolutionary strategy to deform  $C(s, t)$  until an optimum  $C(s)$  is reached.

In [203], each point is attracted towards a local minimum through a gradient descent process, after which, a “basin hopping” operator is applied between local minima (basins) to simplify the energy landscape. If the new local minimum corresponds to a lower energy, then the new solution will be taken without reservation. Otherwise, a coin may be flipped to determine if the new solution is accepted. The updates are global, i.e., instead of each point on the level set function moving in its normal direction at a speed related with the gradient descent, the hopping step is region-based, allowing the search to escape from local minima effectively. In general terms, for a given image  $u_0$ , the piecewise constant Mumford-Shah model [207] seeks for a set of curves  $C$  and a set of constants  $c = (c_1, c_2, \dots, c_n)$  which minimize an energy functional given by

$$F^{MS}(C, c) = \sum_{i=1}^n \int_{\Omega_i} |u_0(x, y) - c_i|^2 dx dy + \mu \times \text{Length}(C) \quad (6)$$

In the latter equation, the curves in  $C$  partition the image into  $n$  mutually exclusive segments  $\Omega_i$  for  $i = 1, 2, \dots, n$ . The idea is to partition the image so that the intensity of each segment  $\Omega_i$  of  $u_0$  is approximated well by a constant  $c_i$ . The goodness-of-fit is measured by the fitting term  $\int_{\Omega_i} |u_0(x, y) - c_i|^2 dx dy$ .

In [201], each agent is affected by ‘forces’ pulling it in a certain direction. These forces are weighted and summed and the resulting vector is normalized to compute a movement potential vector, provided by the function  $P(x)$ . This movement potential function  $P$  is defined as  $P(x) = \|\mathbf{v} + \gamma H(x) + \lambda \|\mathbf{u} - x\|\|$ , where  $x$  is an agent’s location in  $\mathbb{R}^3$ ,  $\mathbf{v}$  is the previous velocity,  $H(x)$  is a steering function moving the agents automatically toward areas of interest (for example, using the gradient  $H$  can lead the agents away from areas characterized by high intensity gradient, i.e., toward more homogeneous areas),  $\mathbf{u}$  is a user-specified location, and  $\gamma$  and  $\lambda$  are constants. In that work, the algorithm regulating an agent’s behavior is entirely deterministic. When the agents are in an area that does not trigger any surface modifications (i.e., when they lie in an image region that does not stimulate them to modify the level set surface), all the agents simply move together along a near-circular path orbiting the user-specified point whilst being pulled toward the path characterized by the lowest image gradient.

#### 4.4.4. Critical Discussion

All papers but [193] are related to medical applications because of the significance and importance that the level set methods have in the segmentation of biomedical images. For instance, some of the approaches discussed in this section have been focused on the segmentation of gray and white matter from simulated T1 magnetic resonance scans [201]; prostate in pelvic computed tomography images [190]; prostate in pelvic computed tomography and magnetic resonance images [191]; lungs in computed tomography and kidneys in



magnetic resonance images [196]; synthetic images, natural scenery, breast tumors and zebrafish cells, and brain in magnetic resonance images [203]; and brain in histological and magnetic resonance images, as well as lung and knee in computed tomography [192, 198, 202].

An interesting option when a set of training images is available, as evidenced by promising results, would be to use an approach similar to those by Ghosh et al. [190, 191] and Mesejo et al. [192]. Nevertheless it is noteworthy that these approaches are quite slow due to the need to compute the average shape and the main forms of variation, as well as the evaluation of texture as the visual feature that characterizes the evolving contour.

Regarding the main roles played by metaheuristics in these methods, interesting working examples can be found which optimize the parameters of a level set model [196, 198], include different terms in the formulation [198], use swarms of agents to refine results [201], take advantage of a training set of shapes to segment a difficult structure [190–192], as well as to solve the Mumford-Shah functional [203] or to quickly initialize the level set [202]. Again, the main advantages of using metaheuristics in combination with level sets derive from the absence of both derivative computation and need to acquire and/or have knowledge about the objective function landscape. The main disadvantages could be the need of a training set to learn the parameters or to extract the main deformation components, and the computation load of some procedures in which fitness calculation is expensive.

The hybridization of level set with metaheuristic appears to be promising for future studies. This happens, firstly, because not all the possibilities have been fully explored up to now. Secondly, because the combination of two techniques with so many positive aspects should lead to results that are more than satisfactory. On the one hand, metaheuristics provide learning and global search capabilities, avoiding local minima, making the initialization of the initial contour robust and the introduction of new fitness terms straightforward, as well as avoiding the need to compute derivatives and to speed up curve evolution. On the other hand, the level set method provides easy management of topological changes and adaptation to solve problems of any dimensions, as well as the opportunity to quickly determine the areas inside and outside the evolving contour.

#### *4.5. Other Approaches*

The metaheuristic-based approaches to deformable models that do not perfectly fit in any of the previous categories have been included in this section. In many cases they could have been relocated in one of the previous sections, but their peculiarities make them worth being treated separately.

Four novel deformable model approaches are presented here: Brownian strings [208, 209], Bayesian dynamic contours [210, 211], adaptive potential active contours [212], and fuzzy active contours [213]. First, the main rationale underlying these approaches will be introduced. The rest of the section has the same structure as

the previous ones, presenting encoding, operators, and fitness functions before closing with a critical discussion.

Unlike snakes and other deformable model approaches, Brownian strings can handle arbitrarily irregular contours in which each interpixel crack (the line segments between pixels are called “crack-edges” in Brownian strings terminology) represents an independent degree of freedom. Thus, Brownian strings are contours described by means of crack-edge chains (see Figure 12) [214]. They rely on a statistically trained external energy term and use simulated annealing to guide contour fitting. The reason why we consider this approach to lie half-way between implicit and explicit deformable models is that it uses two complementary data structures to represent the curve. The first representation is a crack-edge chain, that is an implicit representation of the curve: a sequential enumeration of all vertex coordinates along the curve. In this representation the coordinates of the starting vertex are stored, followed by a 1-D sequence of directional codes (*right*, *up*, *left*, or *down*) denoting the orientations of subsequently traversed cracks. The second data structure is a crack diagram representing the curve as a directed graph linking pixel edges on a rectangular grid. The diagram is implemented as a 2-D crack-edge occupancy array, in which each element can hold at most one crack-edge. The crack diagram is an explicit description of a curve that allows for split/merge-type operations and rapid identification of intersections between contours (see Figure 12).

In Bayesian dynamic contours, each contour is a linked cyclic list of planar nodes, with two sub-models: (i) an a priori contour model which incorporates properties like shape and smoothness, and (ii) an observation-based model accounting for the measured signal intensities. In particular, Bayesian dynamic contours may be considered to belong to the class of active contour models but they differ from the way such models are generally used in three aspects: (a) a wider class of models (energy functions) can be used both for the prior and the likelihood; (b) the optimization technique used is based on stochastic sampling and simulated annealing to provide more flexibility in choosing energy functions; and (c) the number of nodes representing the contour is random (while in the active contour approach is usually fixed, which is too restrictive when more complex contours are to be recognized).

Adaptive potential active contours result from the union of active contour models and classifier construction techniques. Contours are interpreted as contextual pixel classifiers where the context of a pixel consists of the other pixels in its neighborhood. Simulated annealing is used to avoid the local minima of the energy function and the model makes it possible to obtain contours with various topologies. Control points are included and labeled as belonging or not to the object to be segmented. Hence, this is a semi-automatic approach, since the definition of a training set is necessary.

Finally, in fuzzy active contours [213], a fuzzy system tunes the boundaries of the candidate solutions during the refinement phase. It generates a rule-based active contour model approach in which the fuzzy system,

trained by particle swarm optimization, is used to set the weights of the energy terms of a classical snake (internal, external, and thickness terms). Because of this peculiarity, we decided to treat this active contour model separately from the others described in section 4.1.

Table 7: Other approaches distinguishable from the classical ones (snakes, active shape model, level set...). From left to right: author(s) and reference to the paper, deformable model family and model name, dimensions (2-D/3-D), metaheuristics used, image type addressed, and publication year.

Reference	Model Family	Model Name	Dims.	Metaheuristic	Image Modality	Year
Storvik [211]	Parametric	Bayesian dynamic contours	2-D	Simulated Annealing	1 ultrasound and 1 brain magnetic resonance image	1994
Lundervold & Storvik[210]					10 brain magnetic resonance images	1995
Grzeszczuk & Levin[208]	Geometric & Parametric	Brownian strings "Crack-edges"	2-D	Simulated Annealing	1 brain magnetic resonance image, 1 angiogram and 3 synthetic images	1997
Piotrowski & Szczepaniak[209]					Unspecified number of X-ray dental images	2000
Tomczyk [212]	Parametric	Adaptive Potential Active Contours	2-D	Simulated Annealing	2 synthetic images	2007
Wang et al. [213]	Parametric	Fuzzy Active Contour	2-D	Particle Swarm Optimization	15 microscopy images of nerve fibers	2012

#### 4.5.1. Encoding

In [208] and [209], simulated annealing optimizes the positions of the ‘cracks/points’ in the Brownian string that encodes the boundary of the object to be segmented. In Bayesian dynamic contours [210, 211], the genes making up the representation are the cartesian coordinates of every control point in the contour. In turn, adaptive potential active contours [212] try to find the optimal parameter vector of length  $4 \cdot (N_1 + N_2)$  using simulated annealing, where  $N_1$  is the number of control points describing the background and  $N_2$  is the number of control points inside the object. Since contours are interpreted as contextual classifiers of pixels, the search of an optimal contour is considered as a method for constructing optimal classifiers. Finally, in [213], particle swarm optimization is used to derive a fuzzy system by learning the large number of parameters required. This method is divided in three phases, each of which uses a different fuzzy system to: (1) identify nerve fiber candidates (detection phase); (2) set active contour model weights (refinement phase); and (3) eliminate false positives (confirmation phase). The same particle swarm optimization algorithm is used to obtain all the parameters of the three proposed fuzzy rule-based systems.

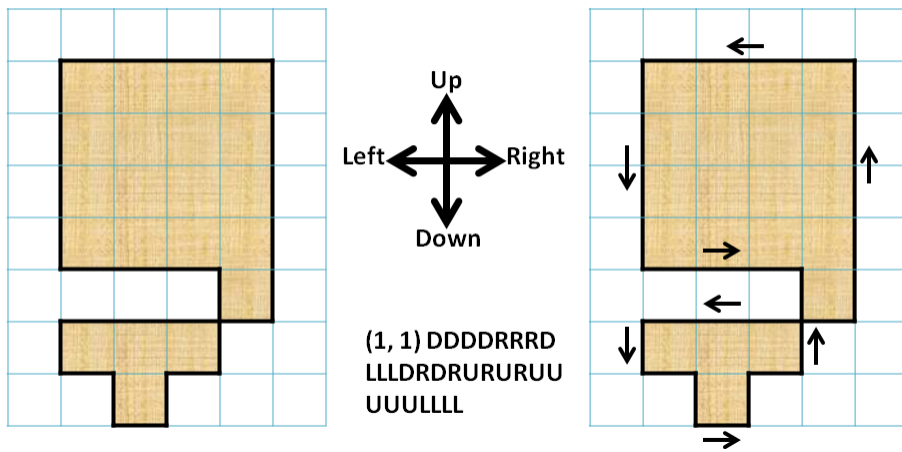


Figure 12: Representation of “Brownian Strings” [214]. A contour is represented by a collection of interpixel “crack-edges”. Every contour (on the left) can be traversed in a counterclockwise fashion, starting at the upper left vertex. The resulting collection of oriented cracks can be represented by a crack-edge chain code (the string in the center), consisting of the coordinates of the starting vertex and a list of the orientations. The oriented cracks in the contour can also be represented explicitly by a 2-D array (on the right of this figure), called a crack diagram.

#### 4.5.2. Operators

In [208–212], simulated annealing is used, usually with an exponential cooling scheme. The formulation of simulated annealing does not favor the development of sophisticated and ad hoc operators. Thus, the methods included here do not boast of presenting complicated operators.

In [208, 209], the authors have developed a “move generator” operator that produces a new state (i.e., a new, slightly perturbed contour configuration). This *ad hoc* procedure must satisfy several stringent requirements. Firstly, all deformations must lead to curves having the required topology (a candidate contour must be closed and must not intersect itself). Secondly, the move generator must be ergodic; i.e., it must be possible to find a series of moves which will deform any curve in the search space into any other curve. Thirdly, the “move” generator should provide the user with a mechanism to control the general size and shape of the deformations. This makes it possible to increase the efficiency of the annealing process by using a spectrum of moves which is appropriate for each temperature level. Finally, it is essential for deformations to be generated in “constant time” (i.e., in a time that is independent of the contour length) in order for the computational burden not to grow rapidly with the contour length. The authors presented two operators satisfying these criteria called “raindrop” move generator and “Bike Trail” method. In each case, the new contour is generated from the previous one via a local perturbation which consists of selecting a segment of the contour and replacing it with a different segment in a way that assures that no self-intersections are formed in the process. In [210, 211], a gradient operator is needed to recognize edges to be included in computing the potential in the energy function.

#### 4.5.3. Fitness Function

In the Brownian approach, the non-parametric energy function is derived from the statistical properties of previously segmented images (training contours). Each contour in the search space is assigned an energy which depends on a global feature that characterizes it. Each crack has an energy that is a function of local image features (e.g., average intensity, gradient, texture, etc.) and the energy of the whole contour is equal to the average energy of all cracks in the contour. During the annealing process, each contour crack is assigned an energy  $E_i = 1 - p$ , where  $p$  is the probability of finding a similar crack in the training contour. In other words, a crack is assigned low (high) energy if similar cracks are found frequently (rarely) in the training contour. Regarding implementation, prior to annealing, the energies of all cracks in the image are precomputed and stored in a 2-D array. Then, during the annealing process, the contour’s energy can be computed efficiently by using this look-up table to find the average energy of all its cracks.

Bayesian dynamic contours use an energy function consisting of internal and external forces where, as usual, the internal forces act as a smoothness constraint and the external ones draw the active contour towards specific image features. Such an energy function is formulated within a Bayesian framework where minimizing the

energy corresponds to finding the maximum a posteriori solution. The prior model captures the knowledge available about the contour (smoothness, curvature, contour length) while the probability distribution of data is computed from the observed image data (gray-levels inside and outside the object, and edges).

The energy of an adaptive potential active contour is composed of an intensity energy term, which attracts the contour towards darker image regions; a homogeneity energy term, which tries to find contours that surround the object composed of pixels with a given intensity; and an internal energy, that evaluates contour shapes. Simulated annealing is used to avoid local minima.

In [213], particle swarm optimization uses the number of correctly classified patterns as a fitness function to optimize all parameters of the three fuzzy rule-based systems. The systems are applied in the three phases (detection, refinement, and confirmation) of the proposed segmentation algorithm. A greedy algorithm is used to minimize the energy of the deformable model, since the search has the features of a local optimization. In fact, initial candidate nerve fibers are obtained by a multi-level watershed scheme and later refined by a fuzzy active contour model, which flexibly deforms contours according to the observed features of each nerve fiber. A final scan with a different set of fuzzy rules removes false detections. No *ad hoc* operators have been developed.

#### 4.5.4. Critical Discussion

In this section, four different approaches have been briefly summarized that are based on principles that differ from the most common ones. In [213], particle swarm optimization is used to create a fuzzy rule-based system that flexibly deforms an active contour model to segment nerve fibers. However, authors use a global-best particle swarm optimization topology, which may be questionable since, usually, the local-best particle swarm optimization using some neighborhood topology obtains better results. In [212], the segmentation process is treated as a classification problem where pixels lying on the contours under consideration are the objects to be classified. This approach tends to be computationally demanding and therefore can be expected to be slow. In [208], an arbitrarily shaped contour, either using an explicit or implicit representation, is stochastically deformed until it fits an object of interest and the non-parametric energy function is derived from the statistical properties of previously segmented images.

Usually, the applicability of these methods has been limited so far to simply connected objects assuming a finite space of possible contours when Bayesian dynamic contours have been used [211] or to very simple synthetic images when adaptive-potential active contours have been used [212]. In many of these approaches, simulated annealing has been the metaheuristic chosen as optimization method. The reason can be found in the fact that simulated annealing is one of the most popular metaheuristics, introduced more than 30 years ago, although more sophisticated methods such as differential evolution or memetic algorithms could also have been used to obtain even better results, or comparable performances in a shorter time.

## 5. Discussion and Recommendations for Future Research

This manuscript is the first attempt to survey examples of hybridization of deformable models and metaheuristics. The only survey partially related to this topic was published in 2009 by Ujjwal Maulik [74], but it deals with all kinds of segmentation techniques, not only deformable models, and it is exclusively focused on the use of genetic algorithms. This section tries to summarize the information provided by the previous sections, identifying some general trends, criticizing them, and trying to derive some general recommendations which may help those who intend to approach segmentation by hybridizing deformable models with metaheuristics.

### 5.1. Design Issues and Recommendations

Regarding the experimentation and the evaluation of results, the use of standard metrics and statistical tests (absent in many works) should be considered from now on as a requirement for a rigorous investigation. Besides, the use and creation of public datasets, and the public dissemination of the results obtained on them would definitely facilitate the comparison between different methods.

Our design recommendation is to use more recent metaheuristics like differential evolution, particle swarm optimization, CMA-ES and scatter search, instead of basic genetic algorithms (based on bit strings, and generally used as off-the-shelf tools, with no attempt to adapt representation, operators or the fitness function more specifically to the problem at hand) or simulated annealing, which are still dominating despite being obsolete or often unable to effectively solve real-world problems [9]. With respect to the use of evolutionary algorithms, operators that best fit the nature of the problem and the characteristics of the representation being used should be considered. For instance, if one is using real-coded genetic algorithms, the best option for crossover would be SBX, PBX or BLX- $\alpha$  [49, 50]. In addition, since fitness computation is the most time-consuming component of a metaheuristic, its effective design and implementation is crucial for a successful application of metaheuristics. Importantly, when tackling an image segmentation problem using energy-minimization techniques like deformable models, the cost function must be defined to be strongly correlated with the resulting segmentation. Otherwise, even the best possible optimizer will lead to suboptimal results.

### 5.2. Possible Directions of Future Research

Very few papers have been published which deal with the automatic tuning of the deformable model parameters. Metaheuristics, or automatic tuners like *irace*[215], can be used to automatically configure computer vision algorithms, instead of relying on costly and inefficient techniques like manual tuning or grid search. This aspect can be considered as a possible direction of future research, as well as a recommendation for researchers in computer vision.

The exploitation of the level set method and prior shape knowledge by means of metaheuristics can be one of the most appealing future trends, due to its actual potential and to the limited number of papers published on this approach up to now [198]. Furthermore, there are very few approaches dealing with local deformations. Usually the deformable model is deformed globally using metaheuristics, when maybe the subdivision of the deformable model into sections to be locally optimized could be an interesting approach.

Finally, especially when dealing with medical image segmentation, it is sometimes possible to define multiple criteria that need to be optimized simultaneously. Hence, another major issue is the application of multi-objective optimization techniques [216, 217] that can be effectively utilized to yield a set of Pareto-optimal solutions from which the domain expert can choose to solve problems that are subject to conflicting constraints [136].

### 5.3. Available Resources

There are many libraries/frameworks containing general-purpose metaheuristic implementations. A potential user can choose from open source libraries, proprietary software, toolboxes in C/C++, MATLAB, CUDA, python or Java. Some libraries/environments that can be of help in solving learning/optimization problems as deformable model design are the following: HeuristicLab [218], Matlab Optimization Toolbox [219], ParadisEO [220], MHTB [221], CILib [222], jMetal [223], JCLEC [224], WEKA [225], libCUDAOptimize [226], Open BEAGLE [227], GALib [228], GAUL [229], MOMH [230], METSlib [231], inspyred [232], and Evoptool [233]. An extensive comparative study of metaheuristic optimization frameworks can be found in [234].

With respect to deformable model implementations, there are also several toolboxes available, mainly in MATLAB and C++, for geometric [235–237] and parametric [238–241] deformable models. Other general computer vision libraries can also help when developing image segmentation algorithms [242–245]. Considering hybridizations of both, to the best of our knowledge, there is only one software package available: the hybridization of differential evolution and active shape models is included within an automatic framework for exploring neurogenomic data and discovering neuropil-enriched RNAs in the hippocampus [246].

## 6. Acknowledgements

This work has been supported by the European Commission with the contract no. 238819 (MIBISOC Marie Curie ITN). Dr. Ibáñez's work has been supported by the Spanish MINECO Juan de la Cierva Fellowship JCI-2012-15359.



## References

- [1] D. L. Pham, C. Xu, J. L. Prince, Current Methods in Medical Image Segmentation, *Annual Review of Biomedical Engineering* 2 (2000) 315–337.
- [2] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis, and Machine Vision*, Thomson-Engineering, 2007.
- [3] D. A. Forsyth, J. Ponce, *Computer Vision A Modern Approach*, Prentice Hall, 2003.
- [4] R. Klette, *Concise Computer Vision - An Introduction into Theory and Algorithms*, Springer, 2014.
- [5] C. A. Floudas, P. M. Pardalos (Eds.), *Encyclopedia of Optimization*, Springer, 2009.
- [6] D.-H. Li, M. Fukushima, On the Global Convergence of the BFGS Method for Nonconvex Unconstrained Optimization Problems, *SIAM Journal on Optimization* 11 (4) (2000) 1054–1064.
- [7] J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Trans. Syst. Man Cybern.* 16 (1) (1986) 122–128.
- [8] V. Zografos, Comparison of Optimisation Algorithms for Deformable Template Matching, in: *Proc. of the International Symposium on Advances in Visual Computing: Part II (ISVC)*, 1097–1108, 2009.
- [9] F. Glover, G. A. Kochenberger (Eds.), *Handbook of metaheuristics*, Kluwer Academic Publishers, 2003.
- [10] D. Terzopoulos, K. Fleischer, Deformable Models, *The Visual Computer* 4 (1988) 306–331.
- [11] D. Terzopoulos, A. Witkin, M. Kass, Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion, *Artificial Intelligence* 36 (1988) 91–123.
- [12] D. Adalsteinsson, D. Adalsteinsson, J. A. Sethian, J. A. Sethian, A Fast Level Set Method for Propagating Interfaces, *Journal of Computational Physics* 118 (1994) 269–277.
- [13] R. Malladi, J. A. Sethian, B. C. Vemuri, Shape modeling with front propagation: a level set approach, *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 17 (2) (1995) 158–175.
- [14] J. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, Cambridge University Press, 1999.
- [15] A. K. Jain, Y. Zhong, M.-P. Dubuisson-Jolly, Deformable template models: a review, *Signal Processing* 71 (2) (1998) 109–129.

- [16] T. Cootes, A. Hill, C.J.Taylor, J.Haslam, M. M. Pt, Use of Active Shape Models For Locating Structures in Medical Images, 1994.
- [17] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, *International Journal of Computer Vision* 1 (1988) 321–331.
- [18] I. Cohen, L. D. Cohen, N. Ayache, Using deformable surfaces to segment 3-D images and infer differential structures, *CVGIP: Image Understanding* 56 (2) (1992) 242–263.
- [19] T. Mcinerney, D. Terzopoulos, Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation, *IEEE Trans. on Medical Imaging* 18 (1999) 840–850.
- [20] J. Montagnat, H. Delingette, N. Ayache, A review of deformable surfaces: topology, geometry and deformation, *Image and Vision Computing* 9 (14) (2001) 1023–1040.
- [21] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active Appearance Models, in: *Proc. of the European Conference on Computer Vision (ECCV)*, vol. 2, 484–498, 1998.
- [22] T. Heimann, H.-P. Meinzer, Statistical shape models for 3D medical image segmentation: a review, *Medical Image Analysis* 13 (2009) 543–563.
- [23] L. He, Z. Peng, B. Everding, X. Wang, C. Y. Han, K. L. Weiss, W. G. Wee, A comparative study of deformable contour methods on medical image segmentation, *Image and Vision Computing* 26 (2008) 141–163.
- [24] C. Xu, D. L. Pham, J. L. Prince, Medical Image Segmentation Using Deformable Models, in: *Handbook of Medical Imaging*, 129–174, 2000.
- [25] M. Bro-Nielsen, Active nets and cubes, IMM Technical Report .
- [26] K. Tsumiyama, K. Yamamoto, Active net: Active net model for region extraction, *IPSI SIG notes* 89(96) .
- [27] N. Barreira, M. Penedo, L. D. Cohen, M. Ortega, Topological active volumes: A topology-adaptive deformable model for volume segmentation, *Pattern Recognition* 43 (1) (2010) 255–266.
- [28] N. Barreira, M. G. Penedo, Topological active volumes, *EURASIP Journal on Advances in Signal Processing* 13 (1) (2005) 1937–1947.
- [29] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, Active shape models-their training and application, *Computer Vision and Image Understanding* 61 (1995) 38–59.

- [30] T. F. Cootes, G. Edwards, C. Taylor, Comparing Active Shape Models with Active Appearance Models, in: *Procs. of British Machine Vision Conference*, 173–182, 1999.
- [31] Y. Zhong, A. K. Jain, Object tracking using deformable templates, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (2000) 544–549.
- [32] A. K. Jain, Y. Zhong, S. Lakshmanan, Object Matching Using Deformable Templates, *IEEE Trans. Pattern Analysis and Machine Intelligence* 18 (1996) 267–278.
- [33] D. Rueckert, Segmentation and Tracking in cardiovascular MR Images using Geometrically Deformable Models and Templates, Ph.D. thesis, University of London, 1997.
- [34] V. Caselles, Geometric models for active contours, in: *Procs. of International Conference on Image Processing*, vol. 3, 1995.
- [35] B. B. Kimia, A. R. Tannenbaum, S. W. Zucker, Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space, *International Journal of Computer Vision* 15 (1994) 189–224.
- [36] R. Kimmel, A. Amir, A. M. Bruckstein, Finding Shortest Paths on Surfaces Using Level Sets Propagation, *IEEE Trans. Pattern Analysis and Machine Intelligence* 17 (6) (1995) 635–640.
- [37] G. Sapiro, A. Tannenbaum, Affine Invariant Scale-Space, *International Journal of Computer Vision* 11 (1) (1993) 25–44.
- [38] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *Journal of computational physics* 79 (1) (1988) 12–49.
- [39] V. Caselles, R. Kimmel, G. Sapiro, Geodesic Active Contours, *International Journal of Computer Vision* 22 (1997) 61–79.
- [40] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, Wiley Publishing, 2009.
- [41] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [42] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, *Science* 220 (1983) 671–680.
- [43] E. Aarts, J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, 1989.

- [44] H. R. Lourenço, O. C. Martin, T. Stützle, Iterated local search, in: Handbook of metaheuristics, Springer, 320–353, 2003.
- [45] P. Hansen, N. Mladenovic, J. A. Moreno-Pérez, Variable neighbourhood search: methods and applications, *Annals of Operations Research* 175 (1) (2010) 367–407.
- [46] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [47] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., 1989.
- [48] Z. Michalewicz, *Genetic algorithms + data structures = Evolution Programs*, Springer-Verlag, ISBN 9783540606765, 1996.
- [49] F. Herrera, M. Lozano, J. L. Verdegay, Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, *Artificial Intelligence Review* 12 (1998) 265–319.
- [50] F. Herrera, M. Lozano, A. M. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study, *International Journal of Intelligent Systems* 18 (2003) 309–338.
- [51] T. Bäck, H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization, *Evol. Comput.* 1 (1) (1993) 1–23.
- [52] H.-P. Schwefel, *Numerical Optimization of Computer Models*, John Wiley & Sons, Inc., 1981.
- [53] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: *Procs. of IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, 1769–1776, 2005.
- [54] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley, 1966.
- [55] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2006.
- [56] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: *Proc. of IEEE International Conference on Neural Networks*, vol. 4, 1942–1948, 1995.
- [57] R. Storn, K. Price, Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, *Tech. Rep.*, International Computer Science Institute, 1995.
- [58] J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in: *Congress on Evolutionary Computation (CEC)*, 1980–1987, 2004.

- [59] D. Zaharie, Influence of Crossover on the Behavior of Differential Evolution Algorithms, *Appl. Soft Comput.* 9 (3) (2009) 1126–1138.
- [60] S. Das, P. Suganthan, Differential Evolution: A Survey of the State-of-the-Art, *IEEE Trans. on Evolutionary Computation* 15 (2011) 4–31.
- [61] P. Moscato, A gentle introduction to memetic algorithms, in: *Handbook of Metaheuristics*, Kluwer Academic Publishers, 105–144, 2003.
- [62] F. Glover, M. Laguna, R. Mart, Scatter search, in: *Advances in Evolutionary Computing: Theory and Applications*, Springer-Verlag, 519–537, 2003.
- [63] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, 2nd edition, Springer Verlag, 2015.
- [64] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, Wiley Publishing, 2nd edn., 2007.
- [65] H. R. Lewis, C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice Hall PTR, 2nd edn., 1997.
- [66] G. Rudolph, Convergence analysis of canonical genetic algorithms, *IEEE Trans. on Neural Networks* 5 (1) (1994) 96–101.
- [67] H.-G. Beyer, *The Theory of Evolution Strategies*, Springer, 2001.
- [68] X.-S. Yang, *Metaheuristic Optimization: Algorithm Analysis and Open Problems*, Springer, 21–32, 2011.
- [69] X.-S. Yang, Theory and New Applications of Swarm Intelligence, chap. *Swarm-Based Metaheuristic Algorithms and No-Free-Lunch Theorems*, InTech, 1–16, 2012.
- [70] M. D. Vose, *The Simple Genetic Algorithm: Foundations and Theory*, MIT Press, 1998.
- [71] W. Gutjahr, Convergence Analysis of Metaheuristics, in: *Metaheuristics*, vol. 10 of *Annals of Information Systems*, Springer US, 159–187, 2010.
- [72] C. Blum, A. Roli, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys* 35 (3) (2003) 268–308.
- [73] S. Cagnoni, Evolutionary Computer Vision: A Taxonomic Tutorial, in: *Procs. of the International Conference on Hybrid Intelligent Systems (HIS)*, 1–6, 2008.
- [74] U. Maulik, Medical Image Segmentation Using Genetic Algorithms, *IEEE Trans. on Information Technology in Biomedicine* 13 (2009) 166–173.

- [75] L. Ballerini, Genetic snakes for medical images segmentation, in: *Procs. of Conference on Mathematical Modeling and Estimation Techniques in Computer Vision*, vol. 3457, 284–295, 1998.
- [76] J. Kidder, B. Nnaji, 3D segmentation of medical images for computer-aided design and rapid prototyping of orthopedic devices, in: Nnaji, BO and Wang, A (Ed.), *Procs. of Conference on Sensors and Controls for Advanced Manufacturing*, vol. 3201, 72–83, 1997.
- [77] L. MacEachern, T. Manku, Genetic algorithms for active contour optimization, in: *Procs of the International Symposium on Circuits and Systems (ISCAS)*, C229–C232, 1998.
- [78] Z. Delu, Z. Zhiheng, X. Shengli, Vector bundle constraint for particle swarm optimization and its application to active contour modeling, *Progress in Natural Science* 17 (10) (2007) 1220–1225, ISSN 1002-0071.
- [79] R. Li, Y. Guo, Y. Xing, M. Li, A Novel Multi-Swarm Particle Swarm Optimization algorithm Applied in Active Contour Model, in: Zhou, SM and Wang, W (Ed.), *Procs. of the Congress on Intelligent Systems*, 139–143, 2009.
- [80] D. Marthaler, A. Bertozzi, Tracking environmental level sets with autonomous vehicles, in: Butenko, S and Murphey, R and Pardalos, PM (Ed.), *Recent Developments in Cooperative Control and Optimization*, vol. 3, 317–332, 2004.
- [81] H. Mobahi, M. N. Ahmadabadi, B. N. Araabi, Swarm contours: A fast self-organization approach for snake initialization, *Complexity* 12 (1) (2006) 41–52.
- [82] E. Shahamatnia, M. Ebadzadeh, Application of particle swarm optimization and snake model hybrid on medical imaging, in: *Procs. of IEEE International Workshop on Computational Intelligence In Medical Imaging*, 1–8, 2011.
- [83] C.-C. Tseng, J.-G. Hsieh, J.-H. Jeng, Active contour model via multi-population particle swarm optimization, *Expert Systems with Applications* 36 (3) (2009) 5348–5352.
- [84] C.-C. Tseng, J.-H. Jeng, J.-G. Hsieh, Active contour model based on multi-population particle swarm optimization, in: *Procs. of International Conference on Systems, Man and Cybernetics*, 2396–2400, 2006.
- [85] K. Wang, Q. Guo, D. Zhuang, H. Chu, B. Fu, Application of snake model based on PSO in the image segmentation, in: *Procs. of Congress on Intelligent Control and Automation*, 9637–9640, 2006.
- [86] X. Wang, Y. Feng, Z. Feng, Ant colony optimization for image segmentation, in: *Procs. of International Conference on Machine Learning and Cybernetics*, 5355–5360, 2005.

- [87] A. C. Cristhian, R. Sánchez, E. Baradit, Detection of knots using X-ray tomographies and deformable contours with simulated annealing, *Wood Research* 53 (2) (2008) 57–66.
- [88] A. C. Cristhian, M. A. Ramos, A. D. Sappa, Simulated Annealing: A Novel Application of Image Processing in the Wood Area (2012) 91–104.
- [89] M. Plissiti, D. Fotiadis, L. Michalis, G. Bozios, An automated method for lumen and media-adventitia border detection in a sequence of IVUS frames, *IEEE Trans. on Information Technology in Biomedicine* 8 (2) (2004) 131–141.
- [90] L. Tang, K. Wang, G. Feng, Y. Li, An image segmentation algorithm based on the simulated annealing and improved snake model, in: Ming, A and Guo, S and Liu, S (Ed.), *Procs. of IEEE International Conference on Mechatronics and Automation*, 3876–3881, 2007.
- [91] D. Chen, Y. Sun, A self-learning segmentation framework - the Taguchi approach, *Computerized Medical Imaging and Graphics* 24 (5) (2000) 283–296.
- [92] J. Tohka, Global Optimization of Deformable Surface Meshes Based on Genetic Algorithms, in: *International Conference on Image Analysis and Processing (ICIAP)*, 459–464, 2001.
- [93] N. Covavisaruch, T. Tanatipanond, Deformable Contour for Brain MR Images by Genetic Algorithm: From Rigid to Training Approaches, in: *Procs. of Image and Vision Computing New Zealand (IVCNZ)*, 7–12, 1999.
- [94] T. Tanatipanond, N. Covavisaruch, A Multiscale Approach to Deformable Contour for Brain MR Images by Genetic Algorithm, in: *Procs. of Symposium on Computational Science and Engineering*, 306–315, 1999.
- [95] Y. Fan, T. Jiang, D. J. Evans, Volumetric Segmentation of Brain Images Using Parallel Genetic Algorithms, *IEEE Trans. on Medical Imaging* 21 (8) (2002) 904–909.
- [96] A. Mishra, P. K. Dutta, M. K. Ghosh, A GA based approach for boundary detection of left ventricle with echocardiographic image sequences, *Image Vision Computation* 21 (11) (2003) 967–976.
- [97] M. Vera, A. Bravo, R. Medina, Myocardial border detection from ventriculograms using support vector machines and real-coded genetic algorithms, *Computers in Biology and Medicine* 40 (4) (2010) 446–455.
- [98] J. Bredno, T. Lehmann, K. Spitzer, Automatic parameter setting for balloon models, in: *Medical Imaging 2000: Image Processing*, vol. 1 of *Procs. of the SPIE*, 1185–1194, 2000.

- [99] S. Cagnoni, A. B. Dobrzeniecki, R. Poli, J. C. Yanch, Genetic algorithm-based interactive segmentation of 3D medical images, *Image and Vision Computing* 17 (1999) 881–895.
- [100] G. M. Teixeira, I. R. Pommeranzbaum, B. L. de Oliveira, M. Lobosco, R. W. dos Santos, Automatic segmentation of cardiac MRI using snakes and genetic algorithms, in: Bubak, M and Dongarra, J and VanAlbada, GD and Sloot, PMA (Ed.), *Computational Science*, vol. 5103 of *LNCS*, 168–177, 2008.
- [101] J. Rousselle, N. Vincent, N. Verbeke, Genetic algorithm to set active contour, in: Petkov, N and Westenberg, MA (Ed.), *Computer Analysis of Images and Patterns*, vol. 2756 of *LNCS*, 345–352, 2003.
- [102] M. Talebi, A. Ayatollahi, Genetic Snake for Medical Ultrasound Image Segmentation, in: Kamel, M and Campilho, A (Ed.), *Image Analysis and Recognition*, vol. 6754 of *LNCS*, 48–58, 2011.
- [103] C.-Y. Hsu, C.-Y. Liu, C.-M. Chen, Automatic segmentation of liver PET images, *Computerized Medical Imaging and Graphics* 32 (7) (2008) 601–610.
- [104] T. Tanatipanond, N. Covavisaruch, An improvement of multiscale approach to deformable contour for brain MR images by genetic algorithm, in: *Procs. of International Symposium on Intelligent Signal Processing and Communication System*, 677–680, 1999b.
- [105] L. Ballerini, Medical image segmentation using genetic snakes, in: Bosacchi, B and Fogel, DB and Bezdek, JC (Ed.), *Procs. of Conference on Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation*, vol. 3812, 13–23, 1999.
- [106] L. Ballerini, An automatic system for the analysis of vascular lesions in retinal images, in: *Procs. of IEEE Nuclear Science Symposium and Medical Imaging Conference*, 1598–1602, 1999.
- [107] L. Ballerini, Detection and quantification of diabetic retinopathy, in: *Procs. of Conference on Applications of Digital Image Processing XXII*, vol. 3808, 213–223, 1999.
- [108] L. Ballerini, Genetic snakes for color images segmentation, in: *Applications of Evolutionary Computing*, vol. 2037 of *LNCS*, Springer Berlin / Heidelberg, 268–277, 2001.
- [109] L. Ballerini, Genetic snakes: active contour models by genetic algorithms, in: *Genetic and Evolutionary Computation in Image Processing and Computer Vision*, *EURASIP Book Series on SP & C*, 177–194, 2007.
- [110] C. Ooi, P. Liatsis, Co-evolutionary-based active contour models in tracking of moving obstacles, in: *International Conference on Advanced Driver Assistance Systems*, 58–62, 2001.



- [111] P. Liatsis, C. Ooi, Tracking moving objects with coevolutionary snakes, in: Grgic, M (Ed.), *Procs.of VIPROMCOM-2002*, 325–332, 2002.
- [112] Y. Xu, S. Neu, C. Ornes, J. Owens, J. Sklansky, D. Valentino, Optimization of active contour model parameters using genetic algorithms: Segmentation of breast lesions in mammograms, in: Sonka, M and Fitzpatrick, JM (Ed.), *Medical Imaging: Image Processing*, vol. 4684 of *Procs. of the SPIE*, 1406–1414, 2002.
- [113] R. Segulier, N. Cladel, Genetic Snakes: Application on Lipreading, in: *International Conference on Artificial Neural Networks and Genetic Algorithms*, 229–233, 2003.
- [114] R. Segulier, N. Cladel, Multiobjectives Genetic Snakes: Application on Audio-Visual Speech Recognition, in: *Procs. of EC-VIP-MC*, vol. 1-2, 625–630, 2003.
- [115] L. Ballerini, L. Bocchi, Bone segmentation using multiple communicating snakes, in: *Medical Imaging: Image Processing*, vol. 5032 of *Procs. of the SPIE*, 1621–1628, 2003.
- [116] L. Ballerini, L. Bocchi, Multiple genetic snakes for bone image segmentation, in: *Applications of Evolutionary Computing*, LNCS, 346–356, 2003.
- [117] L. Ballerini, Multiple genetic snakes for people segmentation in video sequences, in: *Image Analysis*, vol. 2749 of *LNCS*, 275–282, 2003.
- [118] K. Mun, H. Kang, H. Lee, Y. Yoon, C. Lee, J. Park, Active contour model based object contour detection using genetic algorithm with wavelet based image preprocessing, *International Journal of Control, Automation and Systems* 2 (1) (2004) 100–106.
- [119] G. R. Rad, M. Kashanian, Extraction of the breast cancer tumor in mammograms using genetic active contour, in: *Procs. of International Conference on Biomedical and Pharmaceutical Engineering*, 30–33, 2006.
- [120] X. Wu, H. Gao, H. Heo, O. Chae, J. Cho, S. Lee, Y.-K. Lee, Improved B-spline contour fitting using genetic algorithm for the segmentation of dental computerized tomography image sequences, *Journal of Imaging Science and Technology* 51 (4) (2007) 328–336.
- [121] D. Hong, H. You-rui, A novel method with immune genetic algorithm based on snakes for edge detection of concave boundary, in: *Procs. of IEEE International Conference on Control and Automation*, 635–638, 2007.

- [122] A. R. Hussain, Optic nerve head segmentation using genetic active contours, in: *Procs. of International Conference on Computer and Communication Engineering*, 783–787, 2008.
- [123] Z. Chen, B. Zhou, S. Dong, Effective Radical Segmentation of Offline Handwritten Chinese Characters by Using an Enhanced Snake Model and Genetic Algorithm, in: *SAC, Italy*, 54–55, 2012.
- [124] T. Crimmins, A Complete Set of Fourier Descriptors for Two-Dimensional Shapes, *IEEE Trans. on Systems, Man and Cybernetics* 12 (6) (1982) 848–855.
- [125] E. Mezura-Montes, C. A. Coello-Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm and Evolutionary Computation* 1 (4) (2011) 173 – 194.
- [126] J.-M. Yang, C.-Y. Kao, Integrating adaptive mutations and family competition into genetic algorithms as function optimizer, *Soft Comput.* 4 (2) (2000) 89–102.
- [127] J. Paredis, *Coevolutionary Computation*, *Artif. Life* 2 (4) (1995) 355–375.
- [128] K. F. Lai, R. T. Chin, On regularization, formulation and initialization of the active contour models (snakes), in: *Procs. of the Asian Conf. Computer Vision*, 542–545, 1993.
- [129] R. A. Kirsch, Computer determination of the constituent structure of biomedical images, *Computers & Biomedical Research* 4 (1971) 315–328.
- [130] D. Rueckert, P. Burger, A multiscale approach to contour fitting for MR images, in: *Procs. of Conference on Image Processing - Medical Imaging*, vol. 2710, 289–300, 1996.
- [131] N. Bova, O. Cordón, O. Ibáñez, Extended topological active nets, *Image and Vision Computing* 31 (2013) 905–920.
- [132] O. Ibáñez, N. Barreira, J. Santos, M. G. Penedo, Topological active nets optimization using genetic algorithms, in: *Procs. of international conference on Image Analysis and Recognition - Volume Part I (ICIAR)*, 272–282, 2006.
- [133] J. Novo, M. G. Penedo, J. Santos, Localisation of the optic disc by means of GA-optimised Topological Active Nets, *Image Vision Computing* 27 (10) (2009) 1572–1584.
- [134] N. Bova, O. Cordón, O. Ibáñez, Global optimization of Extended Topological Active Nets for image segmentation using Scatter Search, *IEEE Computational Intelligence Magazine* 8 (1) (2013) 16–32.
- [135] O. Ibáñez, N. Barreira, J. Santos, M. G. Penedo, Genetic approaches for topological active nets optimization, *Pattern Recognition* 42 (5) (2009) 907–917.

- [136] J. Novo, J. Santos, M. Penedo, Topological active models optimization with differential evolution, *Expert Systems with Applications* 39 (15) (2012) 12165–12176.
- [137] J. Novo, N. Barreira, M. Gonzalez Penedo, J. Santos, Topological Active Volume 3D segmentation model optimized with genetic approaches, *Natural Computing* 11 (1, SI) (2012) 161–174.
- [138] J. Novo, M. G. Penedo, J. Santos, Evolutionary multiobjective optimization of Topological Active Nets, *Pattern Recogn. Lett.* 31 (13) (2010) 1781–1794.
- [139] J. Novo, J. Santos, M. G. Penedo, A. Fernandez, Optimization of Topological Active Models with Multiobjective Evolutionary Algorithms, in: *Procs. of International Conference on Pattern Recognition (ICPR)*, 2226–2229, 2010.
- [140] J. Novo, J. Santos, M. Penedo, Multiobjective differential evolution in the optimization of topological active models, *Applied Soft Computing* 13 (2013) 3167–3177.
- [141] J. Novo, M. G. Penedo, J. S. Reyes, Multiobjective optimization of the 3D topological active volume segmentation model, in: *Procs. of the International Conference on Agents and Artificial Intelligence (ICAART)*, vol. 1, 236–241, 2011.
- [142] K. Deb, D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, 2001.
- [143] J. Novo, N. Barreira, M. G. Penedo, J. Santos, Genetic Approaches for the Automatic Division of Topological Active Volumes, in: *Procs. of International Work-Conference on The Interplay Between Natural and Artificial Computation (IWINAC)*, 20–29, 2009.
- [144] J. Novo, M. G. Penedo, J. Santos, Optic Disc Segmentation by Means of GA-Optimized Topological Active Nets, in: *Procs. of International conference on Image Analysis and Recognition (ICIAR)*, 807–816, 2008.
- [145] F. M. Ansia, M. Penedo, C. Mario, A. Mosquera, A new approach to active nets, *Pattern Recognition and Image Analysis* 2 (1999) 76–77.
- [146] C. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.
- [147] C. Sierra, J. Novo, J. Santos, M. Penedo, *Frontiers in Artificial Intelligence and Applications*, in: *Advances in knowledge-based and intelligent information and engineering systems*, vol. 243, IOS Press, 1380–1389, 2012.

- [148] C. V Sierra, J. Novo, J. Santos, M. Penedo, Emergent Segmentation of Topological Active Nets by Means of Evolutionary Obtained Artificial Neural Networks, in: *Procs. of the International Conference on Agents and Artificial Intelligence (ICAART)*, vol. 2, 44–50, 2013.
- [149] K. Sakaue, Stereo Matching by the Combination of Genetic Algorithm and Active Net, *Systems and Computers in Japan* 27 (1) (1996) 2239–2246.
- [150] J. Novo, J. Santos, M. G. Penedo, Differential evolution optimization of 3D topological active volumes, in: *Procs. of International conference on Artificial neural networks conference on Advances in computational intelligence (IWANN)*, 282–290, 2011.
- [151] J. Novo, J. Santos, M. G. Penedo, Optimization of Topological Active Nets with Differential Evolution, in: Dobnikar, A and Lotric, U and Ster, B (Ed.), *Adaptive and Natural Computing Algorithms*, vol. 6593 of *LNCS*, 350–360, 2011.
- [152] T. Heimann, S. Münzing, H.-P. Meinzer, I. Wolf, A shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation, in: *Procs. of the International Conference on Information Processing in Medical Imaging (IPMI)*, 1–12, 2007.
- [153] S. Colutto, F. Fruehauf, M. Fuchs, O. Scherzer, The CMA-ES on Riemannian Manifolds to Reconstruct Shapes in 3-D Voxel Images, *IEEE Trans. on Evolutionary Computation* 14 (2) (2010) 227–245.
- [154] C. McIntosh, G. Hamarneh, Medial-based Deformable Models in Non-convex Shape-spaces for Medical Image Segmentation using Genetic Algorithms, *IEEE Trans. on Medical Imaging* 31 (1) (2012) 33–50.
- [155] M. Jolly, S. Lakshmanan, A. Jain, Vehicle segmentation and classification using deformable templates, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18 (3) (1996) 293–308.
- [156] P. Mesejo, R. Ugolotti, S. Cagnoni, F. Di Cunto, M. Giacobini, Automatic Segmentation of Hippocampus in Histological Images of Mouse Brains using Deformable Models and Random Forest, in: *Proc. of Symposium on Computer-Based Medical Systems (CBMS)*, 1–4, 2012.
- [157] P. Mesejo, R. Ugolotti, F. D. Cunto, M. Giacobini, S. Cagnoni, Automatic Hippocampus Localization in Histological Images using Differential Evolution-Based Deformable Models, *Pattern Recognition Letters* 34 (3) (2013) 299 – 307.
- [158] L. Mussi, Š. Ivekovič, S. Cagnoni, Markerless Articulated Human Body Tracking from Multi-View Video with GPU-PSO, in: *Procs. of International Conference on Evolvable Systems: From Biology to Hardware*, 6274, Springer, 195–207, 2010.

- [159] L. Mussi, S. Ivekovic, Y. S. G. Nashed, S. Cagnoni, Multi-View Human Body Pose Estimation with CUDA-PSO, *International Journal of Adaptive, Resilient and Autonomic Systems* 3 (4) (2012) 51–65.
- [160] R. Ugolotti, P. Mesejo, S. Cagnoni, M. Giacobini, F. Di Cunto, Automatic Hippocampus Localization in Histological Images using PSO-Based Deformable Models, in: *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, 487–494, 2011.
- [161] R. Ugolotti, Y. S. Nashed, P. Mesejo, S. Ivekovic, L. Mussi, S. Cagnoni, Particle Swarm Optimization and Differential Evolution for Model-based Object Detection, *Applied Soft Computing* 13 (6) (2013) 3092 – 3105.
- [162] S. Ben Fredj, T. Glatard, C. Casta, P. Clarysse, An Exploration Framework for Segmentation Parameter Spaces, in: *Procs. of International Conference on Image Processing (ICIP)*, 2257–2260, 2011.
- [163] M. Mata, J. M. Armingol, J. Fernandez, A. de la Escalera, Object learning and detection using evolutionary deformable models for mobile robot navigation, *Robotica* 26 (Part 1) (2008) 99–107.
- [164] N. Betrouni, M. Vermandel, J. Rousseau, S. Maouche, Automatic segmentation of prostate boundaries from abdominal ultrasound images using priori knowledge, in: *Procs. of IEEE International Symposium on Biomedical Imaging*, vol. 1-2, 496–499, 2004.
- [165] C. McIntosh, G. Hamarneh, Evolutionary Deformable Models for Medical Image Segmentation: A Genetic Algorithm Approach to Optimizing Learned, Intuitive, and Localized Medial-Based Shape Deformation, in: *Genetic and Evolutionary Computation: Medical Applications*, John Wiley & Sons Ltd, 46–67, 2010.
- [166] M. Mignotte, J. Meunier, J. Tardif, Endocardial boundary estimation and tracking in echocardiographic images using deformable templates and Markov Random Fields, *Pattern Analysis and Applications* 4 (4) (2001) 256–271.
- [167] A. Pitiot, A. Toga, P. Thompson, Adaptive elastic segmentation of brain MRI via shape-model-guided evolutionary programming, *IEEE Trans. on Medical Imaging* 21 (8) (2002) 910 –923.
- [168] D. Rueckert, P. Burger, Geometrically deformable templates for shape-based segmentation and tracking in cardiac MR images, in: *Procs. of International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 1223, 83–98, 1997.

- [169] D. Rueckert, P. Burger, Shape-based segmentation and tracking in 4D cardiac MR images, in: *Procs. of 1st Joint Conference - Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMED-MRCAS)*, vol. 1205 of *LNCS*, 43–52, 1997.
- [170] A. Hill, C. J. Taylor, Model-based image interpretation using genetic algorithms, *Image and Vision Computing* 10 (5) (1992) 295–300.
- [171] A. Hill, C. J. Taylor, T. F. Cootes, Object Recognition by Flexible Template Matching using Genetic Algorithms, in: *Proc. European Conference on Computer Vision (ECCV)*, vol. 588, 852–856, 1992.
- [172] M. Lalonde, L. Gagnon, Variable Neighborhood Search for geometrically deformable templates, in: *Procs. of International Conference on Pattern Recognition (ICPR)*, 689–692, 2002.
- [173] S. Karungaru, M. Fukumi, N. Akamatsu, Feature extraction for face detection and recognition, in: *Procs. of IEEE International Workshop on Robot and Human Interactive Communication*, 235–239, 2004.
- [174] V. Andrey, K.-H. Jo, Road guidance sign recognition in urban areas by structure, in: *Procs. of International Forum on Strategic Technology*, 293–296, 2006.
- [175] A. Sattar, Y. Aidarous, R. Seghier, GAGM-AAM: A Genetic Optimization with Gaussian Mixtures for Active Appearance Models, in: *Procs. of IEEE International Conference on Image Processing (ICIP)*, 3220–3223, 2008.
- [176] A. Sattar, R. Seghier, GGA-AAM: Novel Heuristic Method of Gradient Driven Genetic Algorithm for Active Appearance Models, in: *Procs. of International Conference on Digital Information Management*, 47–52, 2008.
- [177] A. Sattar, R. Seghier, MVAAM (Multi-View Active Appearance Model) Optimized by Multi-Objective Genetic Algorithm, in: *Procs. of IEEE International Conference on Automatic Face and Gesture Recognition*, 96–103, 2008.
- [178] Z. Yin, R. T. Collins, Shape Constrained Figure-Ground Segmentation and Tracking, in: *Procs. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 731–738, 2009.
- [179] H. C. Chen, C. H. Wu, C. J. Lin, Y. H. Liu, Y. N. Sun, Automated Segmentation for Patella from Lateral Knee X-ray Images, in: *Procs. of International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3553–3556, 2009.

- [180] S. A. Szilagy, L. Szilagy, D. Iclanzan, Z. Benyo, A Weighted Patient Specific Electromechanical Model of the Heart, in: *Procs. International Symposium on Applied Computational Intelligence and Informatics*, 101–106, 2009.
- [181] H. Takahashi, M. Komatsu, H. Kim, J. K. Tan, S. Ishikawa, A. Yamamoto, Segmentation Method for Cardiac Region in CT Images Based on Active Shape Model, in: *Procs. of International Conference on Control, Automation and Systems*, 2074–2077, 2010.
- [182] L. Mussi, S. Cagnoni, E. Cardarelli, F. Daolio, P. Medici, P. Porta, GPU implementation of a road sign detector based on particle swarm optimization, *Evolutionary Intelligence* 3 (3-4) (2010) 155–169.
- [183] R. Ugolotti, Y. S. Nashed, S. Cagnoni, Real-Time GPU Based Road Sign Detection and Classification, in: *Procs. of XII Parallel Problem Solving from Nature (PPSN)*, vol. 7491, 153–162, 2012.
- [184] Z. Liu, H. Wang, W. Jiang, Automated tongue area detection for computer-aided diagnosis based on ASM and GA, *Sensor Review* 32 (1) (2012) 39–46.
- [185] P. Fletcher, C. Lu, S. Pizer, S. Joshi, Principal geodesic analysis for the study of nonlinear statistics of shape, *IEEE Trans. on Medical Imaging* 23 (8) (2004) 995–1005.
- [186] S. Pizer, P. Fletcher, S. Joshi, A. Thall, J. Chen, Y. Fridman, D. Fritsch, A. Gash, J. Glotzer, M. Jiroutek, C. Lu, K. Muller, G. Tracton, P. Yushkevich, E. Chaney, Deformable M-Reps for 3D Medical Image Segmentation, *International Journal of Computer Vision* 55 (2-3) (2003) 85–106.
- [187] P. Yip, Y.-H. Pao, Combinatorial optimization with use of guided evolutionary simulated annealing, *IEEE Trans. on Neural Networks* 6 (2) (1995) 290–295.
- [188] J. E. Baker, Reducing Bias and Inefficiency in the Selection Algorithm, in: *Procs. of the 2nd International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, 14–21, 1987.
- [189] J. A. Nelder, R. Mead, A Simplex Method for Function Minimization, *Computer Journal* 7 (1965) 308–313.
- [190] P. Ghosh, M. Mitchell, Segmentation of medical images using a genetic algorithm, in: *Proc. of conference on Genetic and evolutionary computation (GECCO)*, 1171–1178, 2006.
- [191] P. Ghosh, M. Mitchell, J. A. Tanyi, A. Hung, A Genetic Algorithm-Based Level Set Curve Evolution for Prostate Segmentation on Pelvic CT and MRI Images, in: *Biomedical Image Analysis and Machine Learning Technologies: Applications and Techniques*, IGI Global, 127–149, 2010.

- [192] P. Mesejo, S. Cagnoni, A. Costalunga, D. Valeriani, Segmentation of Histological Images using a Metaheuristicbased Level Set Approach, in: *Procs. of Genetic and Evolutionary Computation Conference Companion (GECCO)*, 1455–1462, 2013.
- [193] J. Kan, H. Li, W. Li, Level Set method in standing tree image segmentation based on particle swarm optimization, in: *MIPPR 2007: Automatic Target Recognition and Image Analysis*, vol. 6786 of *Procs. of the SPIE*, V7864, 2007.
- [194] X. Wang, L. He, C. Y. Han, W. Wee, Deformable contour method: A constrained optimization approach, in: *Procs. of the British Machine Vision Conference (BMVC)*, 183–192, 2002.
- [195] M. Xiao, S. Xia, S. Wang, Geometric active contour model with color and intensity priors for medical image segmentation, in: *Procs. of International Conference of the IEEE Engineering in Medicine and Biology Society*, 6496–6499, 2005.
- [196] M. Heydarian, M. Noseworthy, M. Kamath, C. Boylan, W. Poehlman, Optimizing the Level Set Algorithm for Detecting Object Edges in MR and CT Images, *IEEE Trans. on Nuclear Science* 56 (1) (2009) 156–166.
- [197] D. K. Iakovidis, M. A. Savelonas, S. A. Karkanis, D. E. Maroulis, A genetically optimized level set approach to segmentation of thyroid ultrasound images, *Applied Intelligence* 27 (3) (2007) 193–203.
- [198] P. Mesejo, A. Valsecchi, L. Marrakchi-Kacem, S. Cagnoni, S. Damas, Biomedical Image Segmentation using Geometric Deformable Models and Metaheuristics, *Computerized Medical Imaging and Graphics* 43 (2015) 167–178.
- [199] D. Oliveira, R. Feitosa, M. Correia, Genetic Adaptation of Level Sets Parameters for Medical Imaging Segmentation, in: *Biomedical Image Analysis and Machine Learning Technologies: Applications and Techniques*, IGI Global, 150–165, 2010.
- [200] D. Oliveira, R. Feitosa, M. Correia, Segmentation of liver, its vessels and lesions from CT images for surgical planning., *BioMedical Engineering OnLine* 10.
- [201] D. Feltell, L. Bai, 3D level set image segmentation refined by intelligent agent swarm, in: *Procs. of IEEE Congress on Evolutionary Computation (CEC)*, 1–8, 2010.
- [202] P. Mesejo, S. Cagnoni, An experimental study on the automatic segmentation of in situ hybridization-derived images, in: *Proc. on 1st International Conference on Medical Imaging using Bio-Inspired and Soft Computing (MIBISOC)*, 153–160, 2013.



- [203] Y. N. Law, H. K. Lee, A. M. Yip, A Multiresolution Stochastic Level Set Method for Mumford-Shah Image Segmentation, *IEEE Trans. on Image Processing* 17 (12) (2008) 2289–2300.
- [204] X. Wang, L. He, W. Wee, Deformable contour method: A constrained optimization approach, *International Journal of Computer Vision* 59 (1) (2004) 87–108.
- [205] T. W. Ridler, E. S. Calvard, Picture Thresholding Using an Iterative Selection Method, *IEEE Trans. on Systems, Man and Cybernetics* 8 (8) (1978) 630–632.
- [206] C. Li, C. Xu, C. Gui, M. D. Fox, Level Set Evolution without Re-Initialization: A New Variational Formulation, in: *Proc. of the 2006 IEEE Conference on Computer Vision and Pattern Recognition*, 430–436, 2005.
- [207] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Communications on Pure and Applied Mathematics* 42 (5) (1989) 577–685.
- [208] R. Grzeszczuk, D. Levin, “Brownian strings”: Segmenting images with stochastically deformable contours, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19 (10) (1997) 1100–1114.
- [209] M. Piotrowski, P. Szczepaniak, Active contour based segmentation of low-contrast medical images, in: *Procs. of International Conference on Advances in Medical Signal and Information Processing*, 476, 104–109, 2000.
- [210] A. Lundervold, G. Storvik, Segmentation of Brain Parenchyma and Cerebrospinal-fluid in Multispectral MRI, *IEEE Trans. on Medical Imaging* 14 (2) (1995) 339–349.
- [211] G. Storvik, A Bayesian-approach to dynamic contours through stochastic sampling and Simulated Annealing, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 16 (10) (1994) 976–986.
- [212] A. Tomczyk, Image Segmentation Using Adaptive Potential Active Contours, in: Kurzynski, M and Puchala, E and Wozniak, M and Zolnierek, A (Ed.), *Computer Recognition Systems*, vol. 45 of *Advances in Intelligent and Soft Computing*, 148–155, 2007.
- [213] Y.-Y. Wang, Y.-N. Sun, C.-C. K. Lin, M.-S. Ju, Segmentation of nerve fibers using multi-level gradient watershed and fuzzy systems, *Artificial Intelligence in Medicine* 54 (3) (2012) 189–200.
- [214] R. Grzeszczuk, S. Eddins, T. DeFanti, A Region Fill Algorithm Based on Crack Contour Boundaries, *Tech. Rep.*, University of Illinois, 1992.

- [215] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, M. Birattari, The *irace* package, Iterated Race for Automatic Algorithm Configuration, Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
- [216] K. Miettinen, Nonlinear multiobjective optimization, Kluwer Academic Publishers, 1999.
- [217] D. Jones, S. Mirrazavi, M. Tamiz, Multi-objective meta-heuristics: An overview of the current state-of-the-art, *European Journal of Operational Research* 137 (1) (2002) 1 – 9.
- [218] S. Wagner, Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment, Ph.D. thesis, Johannes Kepler University Linz, URL <http://dev.heuristiclab.com/trac/hl/core>, 2009.
- [219] T. Mathworks, Matlab Optimization Toolbox User's Guide, URL [http://www.mathworks.co.uk/access/helpdesk/help/pdf\\_doc/optim/optim\\_tb.pdf](http://www.mathworks.co.uk/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf), 2015.
- [220] S. Cahon, N. Melab, E.-G. Talbi, ParadiseO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics, *Journal of Heuristics* 10 (3) (2004) 357–380, ISSN 1381-1231, URL <http://paradiseo.gforge.inria.fr/>.
- [221] Networking, E. O. group, MetaHeuristics ToolBox for MATLAB, URL <http://neo.lcc.uma.es/software/mhtb/index.php>, 2008.
- [222] G. Pampara, A. P. Engelbrecht, T. Cloete, Cilib: A collaborative framework for Computational Intelligence algorithms - Part I, in: *Procs. of the IEEE World Congress on Computational Intelligence (WCCI)*, 1750–1757, URL <http://www.cilib.net/>, 2008.
- [223] J. J. Durillo, A. J. Nebro, jMetal: A Java framework for multi-objective optimization, *Advances in Engineering Software* 42 (10) (2011) 760–771, URL <http://jmetal.sourceforge.net/>.
- [224] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, C. Hervás, JCLEC: a Java framework for evolutionary computation, *Soft Computing* 12 (4) (2007) 381–392, URL <http://jclec.sourceforge.net/index.php>.
- [225] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA data mining software: an update, *ACM SIGKDD Explorations Newsletter* 11 (1) (2009) 10–18.
- [226] Y. S. Nashed, R. Ugolotti, P. Mesejo, S. Cagnoni, libCudaOptimize: An Open Source Library of GPU-based Metaheuristics, in: *Procs. of the Conference Companion on Genetic and Evolutionary Computation (GECCO)*, 117–124, 2012.

- [227] C. Gagné, M. Parizeau, Open BEAGLE A C++ Framework for your Favorite Evolutionary Algorithm, SIGEvolution 1 (1) (2006) 12–15, URL <https://code.google.com/p/beagle/>.
- [228] M. Wall, GALib: A C++ library of genetic algorithm components, Mechanical Engineering Department, Massachusetts Institute of Technology .
- [229] S. Adcock, Genetic Algorithm Utility Library, URL <http://gaul.sourceforge.net/>, 2009.
- [230] A. Jaskiewicz, G. Dbrowski, MOMH Multiple-Objective MetaHeuristics, URL <http://home.gna.org/momh/index.html>, 2005.
- [231] M. Maischberger, METSlib, URL <https://projects.coin-or.org/metslib>, 2011.
- [232] A. L. Garrett, inspyred: Bio-inspired Algorithms in Python, URL <https://pypi.python.org/pypi/inspyred>, 2012.
- [233] G. Valentini, L. Malagò, M. Matteucci, Evoptool: An extensible toolkit for evolutionary optimization algorithms comparison, in: Procs. of IEEE Congress on Evolutionary Computation (CEC), 1–8, URL [http://airlab.elet.polimi.it/index.php/Evoptool:\\_Evolutionary\\_Optimization\\_Tool](http://airlab.elet.polimi.it/index.php/Evoptool:_Evolutionary_Optimization_Tool), 2010.
- [234] J. A. Parejo, A. Ruiz-Corté, S. Lozano, P. Fernandez, Metaheuristic optimization frameworks: a survey and benchmarking, Soft Computing 16 (3) (2012) 527–561.
- [235] F. Bessy, Ofeli: an Open, Fast and Efficient Level set Implementation, URL <https://code.google.com/p/ofeli/>, 2010.
- [236] G. Papandreou, GAC++: A Toolbox for PDE-Based Image Processing, URL <http://cvsp.cs.ntua.gr/software/GAC++/>, 2008.
- [237] Y. Zhang, 2D/3D image segmentation toolbox, URL <http://uk.mathworks.com/matlabcentral/fileexchange/24998-2d-3d-image-segmentation-toolbox>, 2013.
- [238] D.-J. Kroon, Active Shape Model (ASM) and Active Appearance Model (AAM), URL <http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model--asm--and-active-appearance-model--aam->, 2012.
- [239] D.-J. Kroon, Snake : Active Contour, URL <http://www.mathworks.com/matlabcentral/fileexchange/28149-snake---active-contour>, 2011.

- [240] S. Milborrow, F. Nicolls, Active Shape Models with SIFT Descriptors and MARS, VISAPP URL <http://www.milbo.users.sonic.net/stasm/>.
- [241] M. Luethi, R. Blanc, Stalismo: Framework for building Statistical Image And Shape Models, URL <https://github.com/statismo/statismo>, 2015.
- [242] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, <http://www.vlfeat.org/>, 2008.
- [243] G. Bradski, A. Kaehler, Learning OpenCV, O'Reilly, 2008.
- [244] K. Demaagd, A. Oliver, N. Oostendorp, K. Scott, Practical Computer Vision with SimpleCV, O'Reilly, 2012.
- [245] T. Mathworks, Computer Vision System Toolbox, URL <https://fr.mathworks.com/products/computer-vision/>, 2015.
- [246] R. Ugolotti, P. Mesejo, S. Zongaro, B. Bardoni, G. Berto, F. Bianchi, I. Molineris, M. Giacobini, S. Cagnoni, F. Di Cunto, Visual Search of Neuropil-Enriched RNAs from Brain In Situ Hybridization Data through the Image Analysis Pipeline Hippo-ATESC, PLoS ONE 8 (9) (2013) e74481, URL <http://dx.doi.org/10.1371/journal.pone.0074481>.