Facultade de Informática
Departamento de Computación

PhD Thesis

Topological Active Model Optimization by Means of Evolutionary Methods for Image Segmentation

Author: Jorge Novo Buján
Supervisors: José Santos Reyes
Manuel F. González Penedo
A Coruña, May 2012

To my mother María and brothers Manuel, Ramón, Luís and Julio.
And especially, in memory of my father Ramón.

Acknowledgements

I would like to mention, in the first place, my supervisors José Santos Reyes and Manuel Francisco González Penedo for letting me have the opportunity to do this PhD thesis. Thanks for all the help, suggestions and supervision during these years that have made this work possible.

I especially thank my colleagues Marcos, Noelia, José, Vero, Cas, Marta, Bea and the rest of the members of the VARPA group for all the support, collaborations and shared moments during all this time.

I would also like to express my gratitude to my family and friends. My mother and brothers, thanks for supporting me when I took this path in my career and for being there any time I needed. My godson, my nephew, the rest of my family, my friends of Bandeira and A Coruña, you were always there making everything easier.

Finally, I would like to dedicate this work in memory of my father. You worked so hard all your life to give us the chance to make our dreams come true, but, unfortunately, you cannot be here to enjoy this moment with us. I hope you were proud up there.

Preface

Object localization and segmentation are tasks that have been growing in relevance in the last years. The automatic detection and extraction of possible objects of interest is a important step for a higher level reasoning, like the detection of tumors or other pathologies in medical imaging or the detection of the region of interest in fingerprints or faces for biometrics.

There are many different ways of facing this problem in the literature, but in this Phd thesis we selected a particular deformable model called Topological Active Model. This model was especially designed for 2D and 3D image segmentation. It integrates features of region-based and boundary-based segmentation methods in order to perform a correct segmentation and, this way, fit the contours of the objects and model their inner topology. The main problem is the optimization of the structure to obtain the best possible segmentation. Previous works proposed a greedy local search method that presented different drawbacks, especially with noisy images, situation quite often in image segmentation.

This Phd thesis proposes optimization approaches based on global search methods like evolutionary algorithms, with the aim of overcoming the main drawbacks of the previous local search method, especially with noisy images or rough contours. Moreover, hybrid approaches between the evolutionary methods and the greedy local search were developed to integrate the advantages of both approaches. Additionally, the hybrid combination allows the possibility of topological changes in the segmentation model, providing flexibility to the mesh to perform better adjustments in complex surfaces or also to detect several objects in the scene.

The suitability and accuracy of the proposed model and segmentation methodologies were tested in both synthetic and real images with different levels of complexity. Finally, the proposed evolutionary approaches were applied to a specific task in a real domain: The localization and extraction of the optic disc in retinal images.

Contents

List of Figures

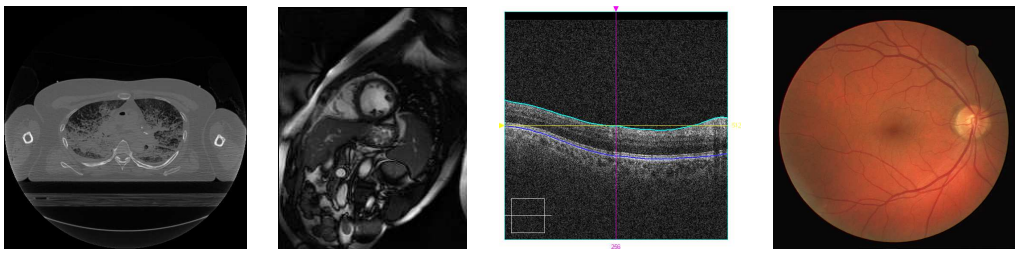
List of Tables

Chapter 1

Introduction

In the recent years, the automatic processing of 2D and 3D image datasets became a relevant task in the way the datasets to be processed and analyzed were extremely increased in many professional areas. For example, in the medical domain, novelty technologies such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), or more specifically in ophthalmology, Optic Coherence Tomography (OCT) or retinal images, provide new information and points of view of real structures of the patient. All this information has to be processed and analyzed by the specialists for medical treatment. Figure 1.1 shows some examples of the different medical images mentioned.

Despite the fact that it is a relatively recent area, the *digital image processing* has demonstrated its great usefulness in the automatic treatment of image datasets. Some of the tasks involved can be, for instance, the morphological image processing, the image recognition or the interpretation systems. Thus, digital image processing can help the specialists to process large datasets, replacing manual procedures which



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

Figure 1.1: Examples of medical images. (a) CT slice. (b) MRI scan: a short axis mid ventricular cardiac magnetic resonance for heart analysis. (c) OCT image. (d) Retinal image.

require long time.

The relevance acquired by the different methods of digital image processing is due to two main areas of application:

Improvement in the image characteristics for human comprehension The aim is the improvement of the quality of the image to be analyzed by a specialist. This kind of techniques are used in different areas, like communication systems or military intelligence.

Processing of the data in a scene The aim is the autonomous interpretation by the machine. That is, the machine can understand what is processing as input information, and to extract the relevant information in the images of interest that corresponds with a real scene and give it a meaning.

Therefore, for this second issue, computer vision techniques can be applied in order to extract and analyze the features of interest in the scene. This issue needs a procedure of different levels of complexity. This implies:

- Detection and segmentation of the objects or regions of interest in the scene.
- Extraction of the characteristics or topological and morphological parameters of these objects or regions.
- Identification and classification of the objects or regions.
- Image understanding of the scene once the objects or regions have been identified.

In this sense, image segmentation and object extraction are crucial tasks in the image understanding process in several domains. One of the most important is medical imaging, where the image understanding is a relevant issue for different tasks: computer-aided diagnosis, surgery planning and simulation or radiotherapy planning, among others. However, these processes normally are complicated because the images are not as ideal as desired. Most of the times, the machines capture the images of interest not perfectly, acquiring the data with different artifacts or noise, making the understanding process more difficult. For that reason, as a crucial task, the segmentation process needs to be as robust as possible, under any possible circumstance.

This thesis proposes new methodologies for object extraction and segmentation, being the main advantage the robustness of the methods and techniques proposed. We used a deformable model called Topological Active Model for the segmentation

and we applied different evolutionary methods for its optimization. This chapter details a general review about the different issues related with such tasks, beginning with an explanation about image segmentation, following with the main principles and works related with deformable models and ending, more specifically, with the related work in the literature that used evolutionary computation to deformable models.

1.1 Image segmentation

As we described before, one of the first steps in the process of image understanding consists of segmenting the image of interest. This segmentation subdivides the image in different parts or regions that constitute the entire scene. The level of subdivision depends on the desired characteristics to be analyzed.

The efficiency of the segmentation is crucial for a process of image understanding, because it is the task responsible of extracting the regions of interest. Due to the data size and the possible variability of the features, the different levels of noise in the images, sampling artifacts or spatial aliasing that can sometimes cause blurred or disconnected boundaries, under all these possible situations the segmentation can be a challenging problem. For that reason, as we explained before, the segmentation technique has to be as much robust as possible to extract the desired objects or regions as better as possible, depending on that the success of any computer vision technique.

The principles of segmentation have their origin in the psychological works of the Gestalt [59], that studied the preferences of the human beings in the organization of groups of shapes in the vision field. These principles indicated that the human being has some specific preferences in the way they organize the perception, based on certain characteristics like *proximity* (objects near among them tend to be grouped together), *similarity* (objects that are similar among them tend to be grouped together) and continuity (objects that conform a close entity tend to be grouped together). This is the way the perception of the human being works. However, these principles are highly difficult to implement in a machine process and different authors in the literature had to develop many different approximations to perform the segmentation.

There are plenty of methodologies developed, based on different principles, that can be applied to 2D and 3D image segmentation. All the methodologies can be organized in different groups depending on different characteristics. For example, based on the image characteristics that are used to detect regions or objects we have

[38]:

Segmentation based on regions The main idea consists of dividing the entire image in different regions, grouping the pixels that belong to one object together. The criteria to group the different points is normally a similarity measure based on properties of a given point and its neighborhood. Examples are clustering methods or region-growing methods.

One of the most used methodologies regarding this category are the *graph cuts* methods [15]. The main idea consists of formulating the segmentation problem as a energy minimization. Representing our problem as a graph, the energy minimization problem can be reduced to instances of the maximum flow problem in the given graph, defining the minimal cut of the graph as well (as detailed by the max-flow min-cut theorem [75]), that represents the best contour of the object to be segmented. There are different versions based on graph cuts, becoming very famous segmentation methods.

Segmentation based on contours or edges They are normally based on the principle that a change in intensity (an edge) points out the separation of two different objects. Thus, the detection of different objects or regions uses the information of edges in the image of interest. An example of this kind of methods are edge detectors. One of the most famous techniques under this category are the *level sets* method. It is explained later in Subsection 1.2.2.

There are different organizations based on different characteristics. Additionally to the previous categorization, one of the simplest and most used classifications is based on the complexity of the method, having that way low-level and high-level techniques [104, 76]. The main principles are:

Low-level techniques These methods are based on simple characteristics of the image, like levels of intensity or the extraction or edges. The simplest are the thresholding methods. They basically turn a grey-scale image into a binary one using a threshold value. There are other methods in this category like histogram based methods or classifiers. The main characteristics of these methods are their simplicity and the efficiency, but they have many problems in the achievement of accurate results in complex images, like noisy images, objects with fuzzy contours or discontinuities, etc.

High-level techniques These methods imply more complex procedures, integrating different characteristics in the images to extract the objects. Moreover,

they normally extract a model related with the objects or regions of interest. As they integrate different features to perform the localization and segmentation, these methods are more robust to the possible complications in the segmentation process and normally obtain better results than the low-level techniques. They also provide structural information of the objects of interest. However, these methods are penalized in efficiency, normally having higher complexities than the low-level ones. Under this category we can find segmentation techniques like deformable models [51], Bayesian methods [79] or atlas-guided techniques [4]. These methods are normally more suitable for image segmentation because they provide better results in complex domains and also provide topological and morphological information about the segmentations achieved.

Regarding image segmentation, one of the most used paradigms is *deformable models*, where is enclosed the proposed work is enclosed. In the next section, we will explain the main basis and related work about the deformable models field.

1.2 Deformable models

One of the most used high-level technique is deformable models. Given the limitations of the low-level techniques, Kass *et al.* proposed a deformable contour in 2D [51] that was also extended to 3D by Terzopoulos *et al.* [97], providing a global solution to the segmentation problem based on the localization of contours. This new way of approaching the image segmentation task implied an improvement with respect to the classic approaches previously defined. Thus, deformable models presented a better way of dealing with the difficult task of segmentation in images with noise or sampling artifacts, or the presence of objects with fuzzy or discontinuous contours. In addition to that, the aim of the segmentation process is not only the extraction of the points that belong to a given object or region, but also to extract its relevant characteristics. Thus, deformable models extract the boundary of the objects and also provide their main features, reconstructing a representation of the structure.

They have been used for several different tasks, like pattern recognition [2, 30], computer animation [95], geometric modelling [47], surgery simulation [29], tracking [13] or image segmentation [50, 67, 108], among others. One of the specific domains where they were widely used is the medical image analysis. Some surveys detail the work related in this field [63, 85].

Deformable models consist of curves, surfaces or solids defined in a given domain (in our case in a given image or set of images). These models are deformed inside

the given domain by a set of forces, both internals and externals. These forces can be described from two different points of view. In a physical point of view, the characteristics of the image determine the external forces, meanwhile the internal forces control the smoothness of the model during the deformations. Moreover, from a mathematical point of view, a deformable model is moved under its dynamic equations, trying to minimize an energy function associated to the model. This energy function represents the correctness of the adjustment, that is, the better the model fits the objects, the lower is the energy.

There is an entire world of deformable models based on different characteristics. In the survey of Montagnat *et al.* [68] the authors explain the main ideas of several deformable models developed. According to their classification, we can organize all the variety of deformable models regarding some relevant characteristics.

1.2.1 Features used as the energy function

As we indicated, the deformable models are deformed according to a set of forces. These forces are represented by different characteristics like image features -for example image intensity or proximity to edges- and model features -like smoothness and contraction-. Most of the deformable models use these forces as a summed energy term that represents the correctness of the adjustment to the objects, that is called the energy function.

One of the first models in the literature was the *snake* [51]. This model is a parametrized model that evolves according to the minimization of an energy function. This energy function includes internal forces, controlling the smoothness and rigidity of the structure, and also external forces, represented by the image features that attract the model to the contour of interest. This approach is simple but robust against noise and missing data. Nevertheless, it is highly dependent on the contour parametrization and the initialization of the model in the image.

After this first approximation, other different models were developed, including different characteristics. For instance, Cohen [24] proposed a model called *balloon*, that is basically an approximation of the model to a sphere by means of polygons. This model can be inflated or deflated until the contour is reaching a contour.

Other different approximations using boundary information are the *geodesic active contours*, proposed by Caselles *et al.* [18] and Kichenassamy *et al.* [52], that are based on a curve evolution approach, opposite to the previous ones that used an energy minimization strategy, and the solution of the model is given by a geodesic curve. As the main drawback, this method has the difficulties to be deformed in complex surfaces like large concavities and, as a solution, Xu and Price developed

a new model called *gradient vector flows* [105], that uses the same principles but it is based on smoothing the gradient field of an image edge map with a non–linear partial differential equation. This way, the attraction range of the image boundaries is extended to the whole image domain.

There is other way of using the image information to perform the segmentation. Thus, instead of using the boundary information, we can use region information. Chan and Vese proposed a new method called *active contours without edges* [19] that mainly performed the minimization of the Mumford-Shad functional [69] with a level set based technique.

Regarding the energy function in all the models, any characteristic with useful information added to the energy function is desired. Thus, if we are working in a given domain, some domain knowledge can be included in the energy function. Thanks to that, the model is more robust to complications in the images, obtaining better results. An example of this kind of models are the *active shape models* [41, 26] and *active appearance models* [32, 25] that use statistical information from a training set of images that contain the features of interest.

1.2.2 Deformable models geometry

One of the main characteristics remains in the fact that a deformable model can be characterized by its surface representation, according to the *shape description* (the model is restricted to represent simple shapes, shapes of restricted topology or shapes with different topologies) and the *deformation description* (by deforming directly the shape or deforming its embedding space, for instance, applying global transformations to the model).

Continuous and discrete representations As a first differentiation regarding geometry, we can have deformable models that present continuous and discrete representations. With a continuous representation we can compute differential quantities, such as surface normals and curvatures, almost everywhere on the surface. Meanwhile, with discrete ones, the surface is only known in a specific set of points. The main problem of the continuous representations is the computational requirements, needing most of the time to be discretized.

Explicit and implicit representations Basically, explicit models are represented by a set of parameters or coordinates, meanwhile implicit models involve an implicit equation to locate the surface points.

Among the explicit models we can find the *polynomial finite support function*,

from where the B-snakes [65], which basically is a B–spline representation of a snake, and its improved version that allows topological changes [56, 55] are the most famous ones. One of the most used ones are the *superquadrics* [94, 8], which basically represents closed surfaces. Such models only allow the representation of symmetric shapes discarding complex surfaces. We also have other types like *deformable templates*, that basically use a set of allowed templates for a specific segmentation, like in the work of Yuille *et al.* [109], or modal decompositions, that use a set of different frequency harmonics [86, 89].

Regarding implicit models, we can find *algebraic surfaces* [92, 93], *superquadrics*, that, in addition to the explicit representation, they can also being formulated in an implicit way [8], or *hyperquadrics*, which is an extension of the superquadric models [23, 22, 39].

Nowadays, one of the most used techniques under implicit representation models is the *level sets* method, technique that was proposed by Osher and Sethian [74] and fully described in [84]. At the first moment, it was proposed for tracking moving interfaces, but then its use spread to other imaging domains, like medical applications [62]. The procedure mainly represents the deformable model as a higher-dimensional scalar function. The contour or surface of the object is the zero-level set of points, where each point is represented as the distance from this point to the higher-dimensional model. Thus, the distance is positive for all the points outside the surface and negative for those that are inside, meanwhile the contour of the segmentation is composed by the set of points with values equal to zero. The evolution of the surface is guided by a partial differential equation regarding the higher-dimensional scalar function that represents the model. The main advantage of this approach is that changes in the topology are allowed easily and implicitly to the model. On the contrary, the main drawback consists of the computational requirements, that is, the numerical solution of the level set equation requires sophisticated techniques, representing the main challenge of the method.

1.2.3 Deformable models evolution

Other relevant aspect of the deformable models is the definition of how the model evolves to reach the desired results. All the deformable models have to be modified to fit the required object, considering the characteristics and restrictions of the model. Most deformable models have an energy function associated. This energy term allows the quantification of the accuracy represented by a given segmentation. This energy term takes low values for correct adjustments meanwhile it increases for worse segmentations. For that reason, we need a mechanism of modification of the

models in terms of reducing the associated energy function.

There are many different techniques to develop this task. One of the easiest way is using a *greedy strategy* like in the work of [103]. The main idea consists of modifying locally the model in a way of reducing the energy until no further modification implies a reduction in terms of energy. As an advantage, these methods are fast in reaching the results, but they are also sensitive to possible noise or complications in the images.

Considering all the complications that can be presented in the datasets, the localization of the global minimum, or at least one acceptable local minimum, is not a trivial issue. There were some approximations like the *Bayesian approach* that uses a statistical framework to do the minimization [88, 109]. Along this line, Terzopoulos and Szeliski [96] depicted a "Kalman snake", based on a probabilistic modelling by adding a Kalman filter to prior models and data with a Bayesian method.

Other global search techniques were developed that become popular and provide acceptable results with a minimum of guarantees. Some of the most used methods are related with the *simulated annealing* [87], *dynamic programming* [3] or *graph cuts* [14], that have been widely used in the literature.

The global search methods minimize the problem of falling in local minima. Among them, we used evolutionary methods as searching algorithms or minimization methods for our deformable model.

1.2.4 Evolutionary approaches and deformable models. Related work

There are few works using evolutionary techniques such as Genetic Algorithms (GA) for the optimization of deformable models. In the case of snake deformable contours, one of the first works which used genetic algorithms was the concept of "genetic snakes" by Ballerini [6, 7], that mainly proposed a GA for the optimization of a snake. Following the same idea, MacEachern and Manku [61] were also among the first to apply GAs for optimizing snakes. Fan et al. [33] applied a parallel GA to optimize active contours in a specific domain, to segment the lateral ventricles from magnetic resonance brain images. Basically, one instance surface was extracted using an active model optimized by finite-differences and used to initialize the first generation of a parallel genetic algorithm that refines the final result. Ooi and Liatsis [72] developed a procedure to perform object tracking in real world scenarios. Different subpopulations of the GA corresponded with different subcontours and all of them evolved in a cooperative manner to achieve the best contour that segmented the object.

Tanatipanond and Covavisaruch [90, 91] also applied a GA to optimize snakes in brain MR images. They used a multiscale approach, beginning at coarser scales to extract rough contours. Then, the best deformable contours at this stage were the parent chromosomes at finer scales. Séguier and Cladel [82] used genetic snakes in a speech recognition application that integrates information from audio and the visual processing of the mouth. In their approach, there were two snakes that define the lips contours and converge in parallel.

Tohka and Mykkänen [100] improved the results of deformable surface meshes by means of a dual contour method in Positron Emission Tomography (PET) brain images. Tohka also used a hybrid approach in [99], where a GA globally minimized the energy of a deformable surface mesh. The minimum obtained was further strengthened by a greedy algorithm. The GA detected roughly the target objects and then the greedy search was used for precise surface extraction.

In other works it was proved the superiority of a global search method by means of a genetic algorithm ([37]) in the optimization of the Topological Active Nets in 2D images ([45, 81]). The results showed that the GA is less sensitive to noise than the greedy algorithm and does not depend on the parameter set or the mesh size.

In the case of 3D segmentation contours, Bro-Nielsen [16] used 3D "active cubes" to segment medical images, where the automatic net division was a key issue. However, they did not use GAs, only an improved greedy algorithm inspired by a simulated annealing procedure to overcome the noise problems.

1.3 Objectives

This PhD thesis proposes new approaches regarding the image segmentation field. Different evolutionary methods are proposed for the optimization of a deformable model called Topological Active Models (TAM). The Topological Active Models are composed by two different versions that were proposed for 2D and 3D images. In particular, two models were developed: The Topological Active Nets (TAN) [101, 16] for segmentation in 2D images and the extension to 3D, the Topological Active Volumes (TAV) [9]. Both models integrate features of region–based and boundary–based segmentation techniques. To this end, the models are defined as squared or volumetric meshes composed by a set of different nodes. The nodes can be of two different types: *external nodes*, with the aim of the extraction of the surface, and *internal nodes*, that model the inner of the objects of interest. The former use boundary information, whereas the latter are related to the region information. The use of both properties presents a good advantage with respect to other approximations to

deformable models because they allow the extraction of the surfaces and also the analysis of the features of the inner of the objects. In both approximations, the nodes are organized as a mesh that has to be deformed under the influence of different functions that conform the energy of the model. Moreover, as other advantage of these deformable models, we can perform topological changes in the model structure to provide flexibility to the mesh, and to perform better adjustments in the presence of holes, concavities or even the segmentation of several objects in the scene.

As explained for other deformable models, the Topological Active Models are governed by an energy function that weights the correctness of the segmentation developed by the model, in such a way that the lower is the energy the better is considered the segmentation performed. As we explained before, the images can present different complications (noise, artifacts captured, boundary discontinuities, etc., ...), that makes the process of segmentation (minimization of the mesh energy) a complex issue.

A previous segmentation technique [9] was developed to minimize the energy using a greedy method. As an advantage, a greedy approach is fast, but it has many problems to reach acceptable results under the detailed complications that can appear in a given dataset. For that reason, global search methods are more suitable for the problem, especially if we focus the properties of the method on the robustness. Thus, in this work, different global search methods using evolutionary approaches were developed trying to overcome the limitations of the previous technique, that is, the greedy local search. Moreover, some hybridizations using the evolutionary global search methods and the greedy local search were performed with the aim of integrate the properties and advantages of both ways of facing the adjustment of the models.

1.3.1 Organization of the thesis

This thesis describes all the evolutionary approaches developed for the image segmentation task, detailing the characteristics of all the methods and the results obtained for each of them.

Chapter 2 discusses all the characteristics of the Topological Active Models (both TAN and TAV models) regarding the topology and the composition of the energy function associated. Moreover, the chapter includes the description of the greedy local search previously developed for the adjustment of the model and also the mechanisms used to perform the changes in the topology.

In chapter 3 the adaptation of a classic GA to the minimization problem of the energy of the model is described. The classic genetic operators were adapted to

the domain and also new *ad hoc* genetic ones were proposed. In the chapter, the entire evolutionary process is described. Moreover, an entire set of experiments are shown to demonstrate the advantages of the global search approach with respect to the greedy local search previously defined. Different difficulties like noise or fuzzy boundaries were used in the experiments. A hybridization of both methodologies is also shown with the corresponding results.

The adapted GA approach presents several limitations. One of them is the slow convergence of the method over the desired results. Chapter 4 describes one of the improvements. A new evolutionary approach, and also the hybridization with the greedy local search, based on Differential Evolution, is used. Representative results are shown and also a comparison between the new evolutionary method and the GA approach was performed, to demonstrate the advantages of the alternative evolutionary method.

One of the important drawbacks of all the single-objective methods, including the proposed evolutionary approaches, in this particular problem, consists of tuning all the energy parameter set. The energy of the model used is defined by different components, weighted by different parameters that measure the relevance of each component in the entire energy function. This parameter tuning has to be developed for each image segmentation, trying to obtain the best possible results in each situation. Chapter 5 explains all the basis of the alternative methodology based on multiobjective optimization. This technique overcomes the limitation of the parameter tuning. Different results and the comparison with the GA approach are shown.

Moreover, in Chapter 6, we include a practical application in which the segmentation techniques developed were used: We used the methods in the task of detecting and segmenting the optic disc in retinal images. A large set of tests was developed to validate and show the robustness of the approximations.

Finally, in Chapter 7 we summarize the main conclusions, together with ideas about possible future research lines related with the work of this thesis.

Chapter 2

Topological Active Models

The Topological Active Net (TAN) [5] and its 3D extension, the Topological Active Volume (TAV) [9] are discrete implementations of an elastic $n-$dimensional mesh with interrelated nodes that integrates features of region–based and boundary–based segmentation techniques, generally called Topological Active Model (TAM).

As other deformable models, the state of the model is governed by an energy function composed by different energy terms related to the mesh nodes. This function weights the correctness of the segmentation so the adjustment process is based on its minimization.

This chapter details the mesh topology and the energy formulation in the TAN model and explains how the 2D model is extended to 3D, emphasizing on the energy terms that represent the objectives to be optimized with the evolutionary approaches. The chapter ends with the description of the initial strategy used for the minimization of the energy of the mesh, a greedy method which incorporates the possibility of changes in the topology of the initial mesh.

2.1 Topological Active Nets

Topological Active Nets are a 2D deformable model based on the Active Nets model proposed by Tsumiyama et al. [101] and developed by Bro-Nielsen [16]. The TAN model shares with those approaches the mesh topology as well as the base energy definitions. However, the TAN model proposes different energy functions and performs different segmentation stages that widen the application fields of the model.

This section explains the bases of the TAN model regarding the initial mesh configuration and the related energy functions.

2.1.1 Topology

A Topological Active Net is a two–dimensional mesh formed by interrelated nodes. The model distinguishes two types of mesh nodes: internal, inside the mesh, and external, on the boundaries. Each type of node represents different features of the objects. The external nodes fit the object surfaces, whereas the internal nodes model the inner object features. This way, the segmentation process integrates boundary and region information in the segmentation process [5, 9]. Figure 2.1 shows a TAN mesh that contains both types of nodes.

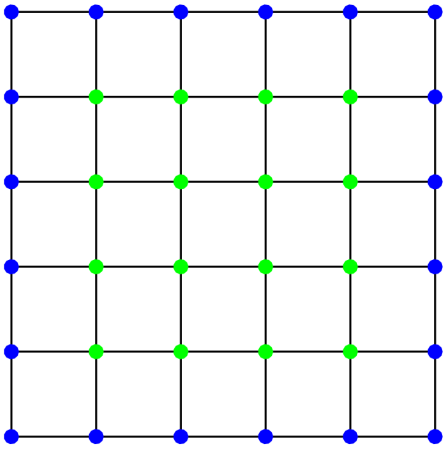


Figure 2.1: A 6×6 TAN mesh. The external nodes are on the boundaries (in blue), whereas the internal nodes are inside the mesh (in green).

As figure 2.1 shows, the TAN nodes are arranged in a polygonal grid formed by squares or rectangles. The edges of these polygons define the neighboring relationships between nodes. Hence, the internal nodes have 4 neighbors, whereas the external ones only have 2 or 3 neighbors regarding their position on the mesh boundaries. The mesh connectivity has influence on the deformation process as well as on the segmentation results, as it will be explained later.

2.1.2 Energies

The TAN model is defined parametrically as $v(r,s) = (x(r,s), y(r,s))$ where $(r,s) \in ([0,1] \times [0,1])$. The mesh deformations are guided by the following energy function:

$$E(v(r,s)) = \int_0^1 \int_0^1 E_{int}(v(r,s)) + E_{ext}(v(r,s))drds \tag{2.1}$$

where E_{int} stands for the internal energy and E_{ext} is the energy due to external forces. The former controls the contraction and bending, whereas the later represents

the features of the scene that guide the adjustment process.

Internal energy objectives The calculus of the internal energy term depends on first and second order derivatives:

$$\begin{aligned} E_{int}(v(r,s)) = \quad & \alpha(|v_r(r,s)|^2 + |v_s(r,s)|^2) \\ + \quad & \beta(|v_{rr}(r,s)|^2 + 2|v_{rs}(r,s)|^2 + |v_{ss}(r,s)|^2) \end{aligned} \tag{2.2}$$

where the subscripts represent the partial derivatives and α, β are weighting terms that control the contraction and bending, respectively. On one hand, large values of α increase the mesh contraction, whereas small values of α restrict the mesh contraction. On the other hand, large values of β produce smooth curves in the mesh, whereas small values of β allow sharp edges.

The definition of the internal energy in 2.2 is continuous. However, the image is a discrete domain. For this reason, the parameter domain $[0,1] \times [0,1]$ is discretized as a regular grid defined by the internode spacing (k, l) and the partial derivatives are computed using the finite difference technique. In this technique, the derivatives are approximated by finite difference formulae based on values of the function at discrete points.

The first order derivatives can be computed using the following approximations:

$$\begin{aligned} d_r^+(r,s) &= \frac{v(r+k,s) - v(r,s)}{k} & d_s^+(r,s) &= \frac{v(r,s+l) - v(r,s)}{l} \\ d_r^o(r,s) &= \frac{v(r+k,s) - v(r-k,s)}{2k} & d_s^o(r,s) &= \frac{v(r,s+l) - v(r,s-l)}{2l} \\ d_r^-(r,s) &= \frac{v(r,s) - v(r-k,s)}{k} & d_s^-(r,s) &= \frac{v(r,s+l) - v(r,s-l)}{l} \end{aligned} \tag{2.3}$$

where d^+, d^o and d^- are the forward, centered and backward differences, respectively.

In the TAN case, the central difference formula cannot be used for the estimation because it only takes into account the contribution of the neighboring nodes. Note that the segmentation process is an energy minimization task and the minimization process is based on changing the position of one or several mesh nodes and comparing the energy values. If the first derivatives are computed by means of the central estimator, when a node changes its position, the first derivative term remains unchanged unless its neighboring nodes have been also updated. Therefore, the central estimator does not allow the evaluation of the quality of the new node position in these cases.

Thus, the first order derivatives are approximated using the backward and for-

ward differences as follows:

$$\begin{aligned} |v_r(r,s)|^2 &= \frac{\|d_r^+(r,s)\|^2 + \|d_r^-(r,s)\|^2}{2} \\ |v_s(r,s)|^2 &= \frac{\|d_s^+(r,s)\|^2 + \|d_s^-(r,s)\|^2}{2} \end{aligned} \tag{2.4}$$

On the contrary, the second order derivatives are estimated using the standard finite difference technique since it takes into account the position of the current node:

$$\begin{aligned} v_{rr} &= \frac{v(r-k,s) - 2v(r,s) + v(r+k,s)}{k^2} \\ v_{ss} &= \frac{v(r,s-l) - 2v(r,s) + v(r,s+l)}{l^2} \\ v_{rs} &= \frac{v(r-k,s) - v(r-k,s+l) - v(r,s) + v(r,s+l)}{kl} \end{aligned} \tag{2.5}$$

External energy objectives While the internal energy term controls the contraction and bending of the TAN, the external energy term represents the features of the scene that guide the adjustment process. This term is defined in such a way that local minima coincide with the image features to segment.

In the TAN model, the external energy of a node is defined as a function of the image intensity and takes into account the neighborhood in the computation. This way, the external energy of a node not only depends on the node itself, but also on the information provided by its neighboring nodes in order to increase the robustness of the segmentation process and avoid local minima caused by noise or intensity inhomogeneities. Hence, the external energy term is defined as follows:

$$E_{ext}(v(r,s)) = \omega f[I(r,s)] + \frac{\rho}{|\aleph(r,s)|} \sum_{p \in \aleph(r,s)} \frac{1}{\|v(r,s) - v(p)\|} f[I(v(p))] \tag{2.6}$$

where ω and ρ are weights, $I(v(r,s))$ is the intensity value of the original image in the position $v(r,s)$, f is a function related to the image intensity, and $\aleph(r,s)$ is the mesh neighborhood of the node (r,s). Thereby, the mesh topology has influence in the energy computation. Note that the contribution of each neighboring node is inversely proportional to its distance to the current node so that the information provided by the closest neighbors has more relevance. This fact ensures a more stable mesh deformation.

The function f reflects the object features that guide the TAN deformation. Since internal and external nodes model different parts of the object, the function f

should be defined in such a way that the internal nodes will remain in the interior of the objects, whereas the external nodes will be attracted by the object boundaries. Also, the color of the objects of interest influences the definition of the function f. So, if the objects are light and the background is dark, the internal nodes should be attracted by light image areas, whereas the external nodes should be attracted by dark areas near the object boundaries. In this case, the function f is defined as follows:

$$f[I(v(r,s))] = \begin{cases} h[I_{max} - \overline{I(v(r,s))_n}] + \tau IOD_i(v(r,s)) & \text{for internal nodes} \\ \\ h[\overline{I(v(r,s))_n} + \xi(G_{max} - G(v(r,s))] & \\ +\varrho GD(v(r,s)) + \tau IOD_e(v(r,s)) & \text{for external nodes} \end{cases} \tag{2.7}$$

where $\overline{I(v(r,s))_n}$ is the mean image intensity in a $n \times n$ window centered at position $v(r,s)$, $G(v(r,s))$ is the gradient image at the position $v(r,s)$, I_{max} and G_{max} are the maximum values of the image intensity and the gradient image respectively. $GD(v(r,s))$ is the gradient distance, i.e., the distance from the node position $v(r,s)$ to its nearest gradient, ξ and ϱ weight the contribution of the gradient image and the gradient distance term and h is an appropriate scaling factor. The term multiplied by h is called In-Out (IO) because it is the energy term that tend to put the external nodes in background intensities and the internal ones in object intensities. Finally, the terms Distance In-Out (IOD), weighted by a factor τ, act as gradients to desirable positions for each kind of nodes: for the internal nodes, $IODi$ points towards values of the object to segment, whereas for the external nodes $IODe$ points towards background intensities. Basically, the IOD energy terms are calculated as the sum of the distances from the position of each node of the mesh to the nearest desirable position, that is, the inner of the object for the internal nodes, and the background for the external nodes, respectively.

The expression $\overline{I(v(r,s))_n}$ is actually a $n \times n$ mean filter so that the input image is smoothed to compute the external energy term. Therefore, the sensitivity to noise is reduced. Also, the function f uses the gradient image and the gradient distance term to check if an external node is over the object boundaries. The minimization of this energy function guides the internal nodes towards the lightest areas of the image, whereas the external nodes are attracted by the darkest pixels near boundaries. For example, in grayscale, the maximum intensity value is 255 and represents a light pixel. Hence, the function f has a minimum for an internal node when the node is on a light pixel ($I_{max} - \overline{I(v(r,s))_n} = 255 - 255 = 0$). On the other hand, the function f has a minimum for an external node when the node is on a dark pixel and near boundaries ($I(v(r,s))_n + \xi(G_{max} - G(v(r,s)) = 0 + \xi(255 - 255) = 0 + 0 = 0$).

On the contrary, if the objects to detect are dark and the background is light, the function f is defined as follows:

$$f[I(v(r,s))] = \begin{cases} h[\overline{I(v(r,s))_n}] + \tau IOD_i(v(r,s)) & \text{for internal nodes} \\ \\ h[I_{max} - \overline{I(v(r,s))_n} + \xi(G_{max} - G(v(r,s))] & \text{for external nodes} \\ +\varrho GD(v(r,s)) + \tau IOD_e(v(r,s)) & \end{cases} \tag{2.8}$$

where the symbols have the same meaning as in equation 2.7. Basically, the main difference is related to the term called IO.

This combination of gradient terms and intensity values in an area allows the integration of both boundary and region information in the external energy term.

2.2 Topological Active Volumes

The objective of the TAN meshes is the 2D image segmentation, that is, the identification of the object boundaries by means of the external nodes and the definition of the interior of the object by means of the internal nodes. In 3D, the segmentation becomes a volumetric task so that the external nodes should fit the object surfaces, whereas the internal nodes should be enclosed inside the object.

The 3D extension of the TAN model is practically straightforward, although there are several issues regarding the mesh topology and the calculus of the energy functions as it will be explained next.

2.2.1 Cubic topology

The cubic topology is the straightforward extension of a 2D topology formed by squares or rectangles [10, 9]. Figure 2.2 shows a $4 \times 4 \times 3$ TAV with cubic topology. In 3D, the external nodes are on the mesh surface, whereas the internal holes are inside the mesh.

The cubic mesh defines a neighborhood where an internal node has 6 neighbors, whereas an external node has 3, 4 or 5 neighbors according to its location on the corner, edge or face of the mesh, respectively.

2.2.2 Energies

Independently of the mesh topology, the TAV model is defined parametrically as $v(r,s,t) = (x(r,s,t), y(r,s,t), z(r,s,t))$ where $(r,s,t) \in ([0,1] \times [0,1] \times [0,1])$. The state of the model is governed by an energy function defined as follows:

$$E(v(r,s,t)) = \int_0^1 \int_0^1 \int_0^1 E_{int}(v(r,s,t)) + E_{ext}(v(r,s,t)) dr ds dt \tag{2.9}$$

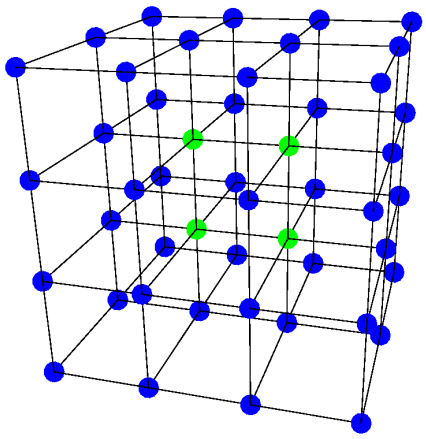


Figure 2.2: A $4 \times 4 \times 3$ cubic mesh. The blue nodes represent the external nodes and the green nodes are the internal nodes.

where E_{int} stands for the internal energy and E_{ext} is the external energy.

Internal energy objectives Like the 2D model, the internal energy controls the shape and the structure of the net and its calculus depends on first and second order derivatives, which control contraction and bending, respectively. In 3D, the internal energy is defined by:

$$\begin{aligned} E_{int}(v(r,s,t)) &= \alpha(|v_r(r,s,t)|^2 + |v_s(r,s,t)|^2 + |v_t(r,s,t)|^2) + \\ &\quad \beta(|v_{rr}(r,s,t)|^2 + |v_{ss}(r,s,t)|^2 + |v_{tt}(r,s,t)|^2) + \\ &\quad 2\gamma(|v_{rs}(r,s,t)|^2 + |v_{rt}(r,s,t)|^2 + |v_{st}(r,s,t)|^2) \end{aligned} \tag{2.10}$$

where subscripts represent partial derivatives and α, β and γ are coefficients that control the smoothness of the mesh.

In order to compute the energy, the parameter domain $[0,1] \times [0,1] \times [0,1]$ is discretized as a regular grid defined by the internode spacing (k, l, m) and the first and second derivatives are estimated using the finite difference technique in 3D. Specifically, the first order derivatives are computed using the forward and backward differences as follows:

$$\begin{aligned} |v_r|^2 &= \frac{\|d_r^+\|^2 + \|d_r^-\|^2}{2} \\ |v_s|^2 &= \frac{\|d_s^+\|^2 + \|d_s^-\|^2}{2} \\ |v_t|^2 &= \frac{\|d_t^+\|^2 + \|d_t^-\|^2}{2} \end{aligned} \tag{2.11}$$

where d^+ and d^- are defined by:

$$\begin{aligned} d_r^+ &= \frac{v(r+k,s,t)-v(r,s,t)}{k} & d_r^- &= \frac{v(r,s,t)-v(r-k,s,t)}{k} \\ d_s^+ &= \frac{v(r,s+l,t)-v(r,s,t)}{l} & d_s^- &= \frac{v(r,s,t)-v(r,s-l,t)}{l} \\ d_t^+ &= \frac{v(r,s,t+m)-v(r,s,t)}{m} & d_t^- &= \frac{v(r,s,t)-v(r,s,t-m)}{m} \end{aligned} \tag{2.12}$$

The finite difference technique is also used for computing the second order derivatives:

$$\begin{aligned} v_{rr} &= \frac{v(r-k,s,t)-2v(r,s,t)+v(r+k,s,t)}{k^2} \\ v_{ss} &= \frac{v(r,s-l,t)-2v(r,s,t)+v(r,s+l,t)}{l^2} \\ v_{tt} &= \frac{v(r,s,t-m)-2v(r,s,t)+v(r,s,t+m)}{m^2} \\ v_{rs} &= \frac{v(r-k,s,t)+v(r,s+l,t)-v(r-k,s+l,t)-v(r,s,t)}{kl} \\ v_{rt} &= \frac{v(r-k,s,t)+v(r,s,t+m)-v(r-k,s,t+m)-v(r,s,t)}{km} \\ v_{st} &= \frac{v(r,s-l,t)+v(r,s,t+m)-v(r,s-l,t+m)-v(r,s,t)}{lm} \end{aligned} \tag{2.13}$$

External energy objectives E_{ext} represents the features of the scene that guide the adjustment process and is defined as follows:

$$\begin{aligned} E_{ext}(v(r,s,t)) = \; & \omega f[I(v(r,s,t))]+ \\ & \frac{\rho}{|\aleph(r,s,t)|}\sum_{p\in\aleph(r,s,t)}\frac{1}{\|v(r,s,t)-v(p)\|}f[I(v(p))] \end{aligned} \tag{2.14}$$

where ω and ρ are weights, $I(v(r,s,t))$ is the intensity value of the original image in the position $v(r,s,t)$, f is a function related to the image intensity, and $\aleph(r,s,t)$ is the neighborhood of the node (r,s,t). Thus, given that the repeated polyhedron in the mesh defines the node neighborhood, the shape of the polyhedron influences not only the flexibility of the mesh, but also the way the nodes are adjusted to the objects.

The function f is defined exactly as in the 2D case; when the objects are light and the background dark:

$$f[I(v(r,s,t))] = \begin{cases} h[I_{max} - \overline{I(v(r,s,t))_n}] \ + \ \tau IOD_i(v(r,s,t)) & \text{for internal nodes} \\ \\ h[\overline{I(v(r,s,t))_n} + \xi(G_{max} - G(v(r,s,t)))] & \\ +\varrho GD(v(r,s,t)) \ + \ \tau IOD_e(v(r,s,t)) & \text{for external nodes} \end{cases} \tag{2.15}$$

where, again: ξ,ϱ are weighting terms, I_{max} and G_{max} are the maximum intensity values of image I and the gradient image G, respectively, $I(v(r,s,t))$ and $G(v(r,s,t))$ are the intensity values of the original image and the gradient image in the node position $v(r,s,t)$, $\overline{I(v(r,s,t))}_n$ is the mean intensity in a $n \times n \times n$ voxel neighborhood, h is an appropriate scaling function multiplying once again the energy term IO, and $GD(v(r,s,t))$ is the gradient distance, i.e., the distance from the node position $v(r,s,t)$ to its nearest edge, and the components IOD are defined as in the 2D case.

Otherwise, if the objects to detect are dark and the background is light, the energy of an internal node will be minimum when it is on a point with a low grey level. On the other hand, the energy of an external node will be minimum when it is on a discontinuity and on a light point outside the object. In this situation, the function f is defined as:

$$f[I(v(r,s,t))] = \begin{cases} h[\overline{I(v(r,s,t))_n}] \ + \ \tau IOD_i(v(r,s,t)) & \text{for internal nodes} \\ \\ h[I_{max} - \overline{I(v(r,s,t))_n} + \xi(G_{max} - G(v(r,s,t)))] + & \\ \varrho GD(v(r,s,t)) \ + \ \tau IOD_e(v(r,s,t)) & \text{for external nodes} \end{cases} \tag{2.16}$$

where the symbols have the same meaning as in equation 2.15.

2.3 Greedy methodology

Most of the minimization techniques are based on performing several steps with a set of choices in each step. The *gradient* methods [71], the *dynamic programming* [12] or the *simulated annealing* [53] techniques are examples of algorithms that explore the search space iteratively. In this sense, a *Greedy algorithm* selects the best local solution at each step of the minimization process.

The local choices provide a compromise that produces acceptable approximations and, sometimes, leads to the global minimum solution [27].

A greedy algorithm is a simple minimization method, usually quite efficient and suitable for problems where the choice at every step can produce an optimal solution, i.e., a solution that minimizes the objective function.

The greedy optimization for the topological model is automatic, so a first step is to establish a suitable initialization of the mesh in order to avoid any human interaction as other deformable models. The greedy strategy implies that successive local

optimizations should converge to an optimal solution. Also, since the external nodes should be on the surfaces, whereas the internal nodes should be inside the objects, a simple initialization strategy is to create a mesh with a homogeneous distribution of nodes within the limits of the whole image. This way, the minimization process will shrink the mesh and will lead the nodes towards the objects of interest.

Then, the energy of the model is minimized using the greedy strategy. At each step, this algorithm tries to minimize the energy functions locally. The mesh energy is computed as the sum of the node energies. Since the greedy strategy performs a local search, the node energy is computed in the current node position within the image and in its neighboring positions. After that, the image position with the lowest energy value is selected as the next node position. The new mesh energy is the sum of the lowest local node energies. These steps are repeated until the mesh energy remains unchanged.

2.3.1 Topological changes. Link cutting procedure

The size and shape of the mesh are established at the beginning of the segmentation process. However, the shape of the object could vary drastically and the final segmentation results could not be as good as possible. Topological changes could be performed in particular areas of the shape with special concavities or irregularities where a better segmentation can be obtained.

The greedy method introduces a mechanism called *link cutting procedure* that is applied to the model when the segmentation cannot be improved. It is based on removing a connection between two wrongly located nodes in the mesh by a previous identification of the external nodes wrongly located. Hence, the flexibility of the mesh in this areas will be increased, and the net will be able to improve the adjustment. These nodes wrongly located are the nodes more distant to the object edges. To this aim we use the Tchebycheff's theorem. This way, an external node v_{ext} is wrongly placed if its gradient distance fulfills the following inequality:

$$GD(v_{ext}) > \mu_{GD} + 3\sigma_{GD} \tag{2.17}$$

where μ_{GD} is the average gradient distance of the whole set of external nodes and σ_{GD} is their standard deviation.

After the identification of the outlier set, the link to remove is selected. It is the link between the node with the highest gradient distance and its worst neighbor in the outlier set. Once the link is cut, some internal nodes become external since they are on the boundaries of the mesh. The increase of the number of external nodes allows a better adjustment to the object boundaries. An example of the breaking

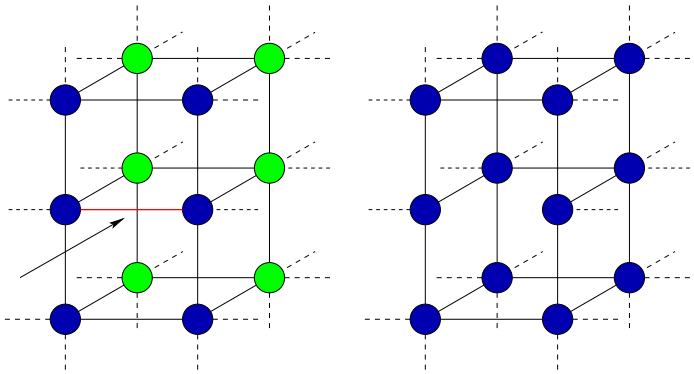


Figure 2.3: 3D link cutting example. Conversion of internal nodes in external ones. (a) Before the breaking and connection to be removed. (b) Result after the breaking. Green and blue nodes represent the internal and the external nodes, respectively.

of a connection is shown in Figure 2.3, where some internal nodes become external ones after the breaking.

The process of breaking implies a restriction: the squared structure in 2D and the cubic one in 3D of the model has to be preserved, that is, all the nodes and connections has to perform squares or cubes, respectively. The situation of nodes and connections isolated is not allowed. Under this circumstance, sometimes a breaking of a connection implies breaking the topology of the model, that is leaving connections and nodes isolated, but several simultaneous breaking can preserve the topology of the mesh. For example, in Figure 2.4 we can see an example of a breaking that implies the breaking of other connections.

It also can be used to identify internal nodes wrong located. That could be possible in the case the object has a hole inside of the object. Using the same mechanism we can remove some links to create an internal hole and obtain a better adjustment in this part of the object. This mechanism detailed in [11], was not used in the current work, where we only used the link cutting procedure in the external nodes, together with the automatic net division explained next.

2.3.2 Automatic mesh division

Since the link cutting process breaks the topology to improve the adjustment, when the image has several objects the net should be divided to segment them. To this end, a reconfiguration mechanism was developed in order to perform multiple object detection and segmentation.

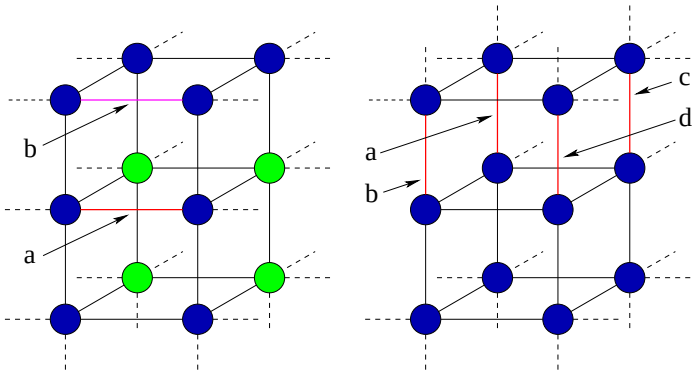


Figure 2.4: 3D multiple breaking example. The marked connections has to be removed because: (a) The breaking of connection a implies that connection b does not belong to any cube. (b) If any of these connection is removed, the other has to be removed as well, because they do not belong to any cube anymore.

The mesh division is performed by the link cutting procedure. However, this algorithm cannot be applied directly to the automatic division. Since the topology must be preserved, problems arise when cutting a link implies leaving isolated links or planes. In such case, these links cannot be cut so a "thread" composed by squares in 2D or cubes in 3D will appear between two submodels. If one connection in one of these squares or cubes is broken, the topology is not preserved. Figure 2.5 shows these ideas in 2D for a better visualization. Figure 2.5(a) presents an example with a "thread". Figure 2.5(b) depicts a case that leads to threads. If the labeled link is removed, there will be two threads since no other link can be cut. The 3D case is equivalent.

However, this problem can be overcome if we consider a direction in the cutting process [16]. Thus, a cutting priority is associated to each node whose connection is removed. A higher priority is assigned to the nodes in the cutting direction whereas a lower priority is assigned to the nodes involved in the cut. Figure 2.5(c) shows the recomputation of the node priorities after several cuts in the 2D case. The extension for the 3D case is straightforward. Figure 2.6 also shows a 3D example about how to deal with the link cutting procedure, using priority or not. As we can see, the version that uses priority is able to divide the mesh (Figure 2.6 (d)), meanwhile the one without priority gets stacked leaving threads between both sub-meshes (Figure 2.6 (c)).

The cutting priority weights the gradient distance of each node. Thus, once the set of badly placed external nodes is obtained using equation 2.17, the link to be

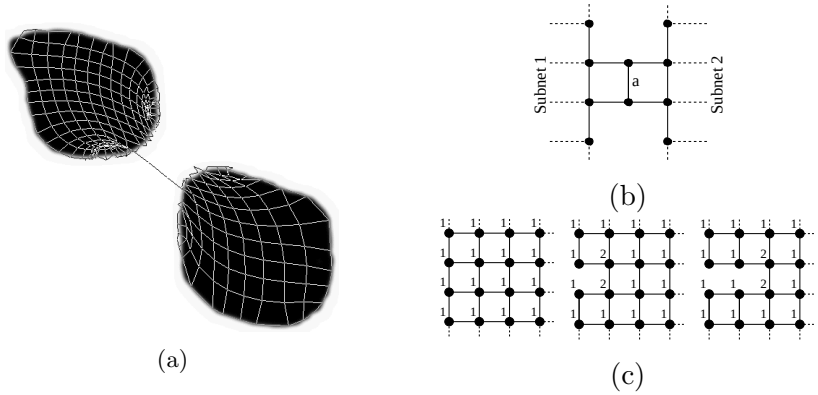


Figure 2.5: Threads and cutting priorities in 2D. (a) Image segmentation with threads. (b) If link "a" is removed, no other link can be removed in order to preserve the TAN topology. (c) Recomputation of cutting priorities. When a link is broken in a direction, the neighborhood in this direction increases its priorities.

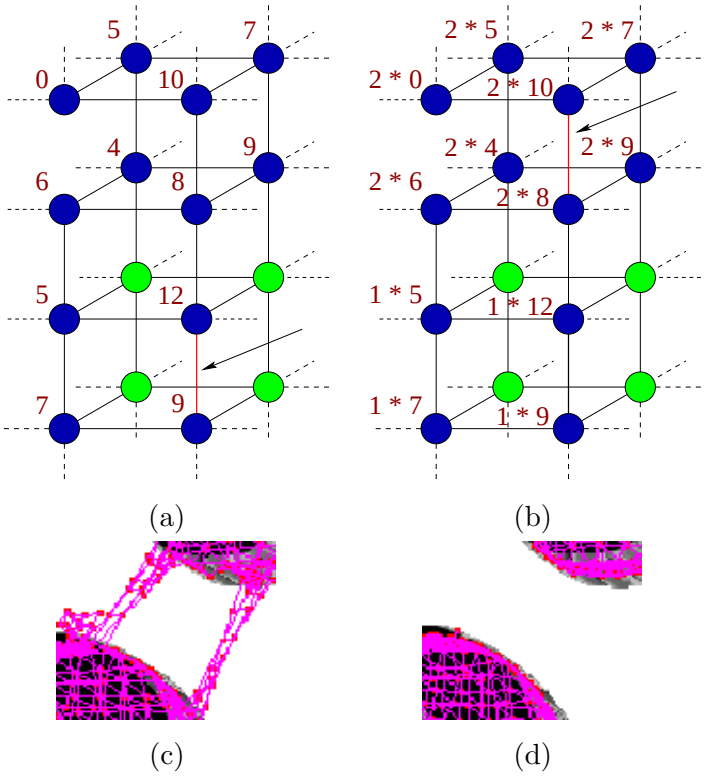


Figure 2.6: Threads and cutting priorities in 3D. (a) (b) Next connections to be removed without priority and with it, respectively. (c) (d) Examples of final results without using priority and with it.

removed consist of two neighboring nodes within this set, n_1 and n_2, that fulfill:

$$\begin{aligned} &GD_{v_{ext}}(n_1) \times P_{cut}(n_1) > GD(n) \times P_{cut}(n),\ \forall n \neq n_1 \\ &GD_{v_{ext}}(n_2) \times P_{cut}(n_2) > GD_{v_{ext}}(m) \times P_{cut}(m),\ \forall m \neq n_2, m \in \aleph(n_1), \end{aligned} \tag{2.18}$$

where $P_{cut}(x)$ is the cutting priority of node x, $GD_{v_{ext}}(x)$ is the distance from the position of the external node x to the nearest edge, and $\aleph(n_1)$ is the set of neighboring nodes of n_1.

Summarizing, the entire segmentation process including the possible topological changes can be seen in Figure 2.7.

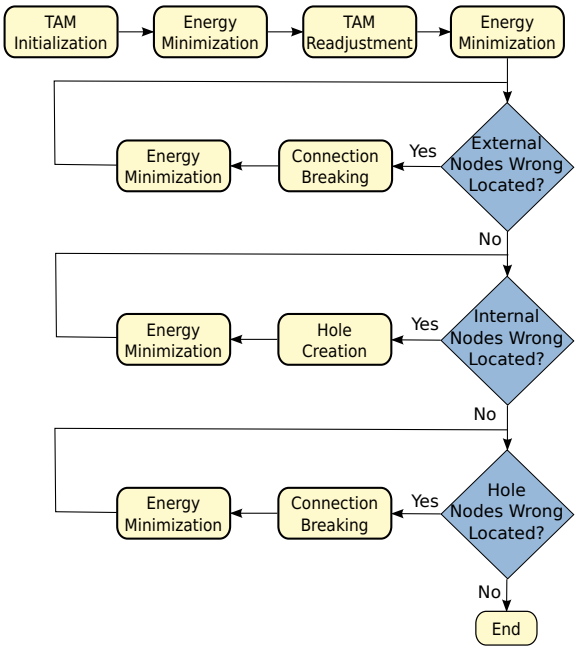


Figure 2.7: Segmentation process using a greedy strategy diagram.

2.3.3 Advantages and disadvantages

This minimization technique has as the main advantage the speed in the segmentation process because it moves progressively the nodes in the neighboring pixels while the energy is decreased, until no further energy minimization is possible. This way, the method is fast and direct, and it is capable to reach good results in low time under good image conditions. Moreover, the introduction of topological changes is easy. When no further movement in all the nodes can be done, we can analyze all the links to locate which ones are not well placed.

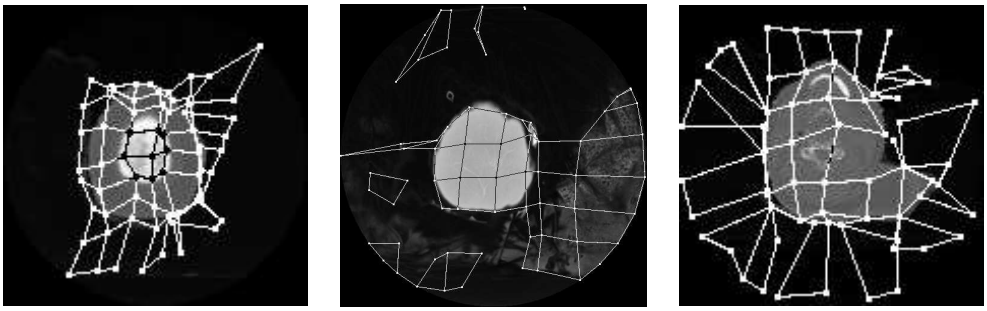


Figure 2.8: Wrong 2D segmentations using the greedy methodology.

However, this method has difficulties in providing acceptable results under complex conditions, like noisy images or objects with fuzzy or discontinuous contours. As the method moves the nodes locally to the best neighboring pixel, it is easy to get stacked in noisy regions of the images, falling in local minima. Figure 2.8 shows some results obtained with the greedy local search in some real CT images that presented some artifacts captured that cannot be overcome by the method. The results are not acceptable.

These limitations motivated the search of more robust methodologies that can reach acceptable results under any possible complication in the images, situation quite common in many image processing areas, like the medical domain. Evolutionary computation demonstrated to be suitable global search methods to find acceptable results working in very complex optimization problems.

Chapter 3

Optimization of Topological Active Models by means of Genetic Algorithms

3.1 Introduction

The greedy strategy is a deterministic local search method. Beginning from the same conditions, the method always converge to the same results. The main advantage is that the method provides a result quite fast and direct and requires low memory. However, the method is not capable to reach acceptable results in complex and noisy images. Most of the possible situations in a real domain presents this kind of conditions.

Given the variability of the objects and conditions in the images, it is recommended to use a global search method to be able to reach global correct solutions in all the possible situations. There are several global minimization approaches in literature such as the *branch and bound* algorithms [40], *simulated annealing* [53], *tabu search* [36] or *evolutionary strategies* [37]. The *branch and bound* algorithms consist of a systematic enumeration of the whole set of candidate solutions by using super and lower estimated bounds of the quantity to optimize. Typically, these techniques rely on some *a priori* structural knowledge about the problem. The *simulated annealing* approach considers, at each step, some neighbors of the current state and decides, given a probability, if the system is moved to the new state or not. It allows non optimal states in the search. This strategy finds a good approximation of the global minimum, but it does not guarantee the global optimum. The *tabu search* forbids states already visited in the search space at least for the upcoming few events.

It is similar to the *simulated annealing* since it accepts new inferior solutions temporarily to avoid paths already investigated. Nevertheless, the *tabu search* selects new states based on their quality, not at random like the previous technique. The *evolutionary techniques* are based on biological evolution and natural selection laws. The candidate solutions are coded as a population of individuals that are combined and mutated into new individuals (solutions) whose quality is evaluated by means of a fitness function.

In the TAM minimization problem, given that there is no *a priori* knowledge about the minimum of the energy functions, the use of a *branch and bound* algorithms is not suitable. Moreover, other optimization techniques such as the *simulated annealing* or the *tabu search* are based on a probabilistic local search and could not reach the global minimum. Therefore, an evolutionary technique, with a simultaneous search over the points of the search space represented by the individuals of the population, may be a complementary approach to find the global minimum in the segmentation process we are dealing with.

3.2 Adapted Genetic Algorithm

Genetic Algorithms (GA) [42, 37] are a particular type of evolutionary techniques broadly applied to optimization problems. The use of a GA in a minimization process requires the definition of a set of characteristics that are defined below.

Genotypic encoding The genetic algorithms require that the set of variables or parameters to be optimized in a specific problem have to be coded in a chromosome. Each chromosome, genotype or individual represents a candidate solution of the problem and consists of a set of genes that encodes the optimization parameters. Thus, we can represent possible solutions in the entire search space as individuals that compose a genetic population. In our method, each genotype is represented by the list of the coordinates of all the nodes of the mesh. Thus, the chromosome is composed by a list of integers:

$$x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n \tag{3.1}$$

where n is the number of nodes in the mesh, and z is the third coordinate only considered in 3D segmentations.

Fitness function We also need the definition of a fitness function to evaluate the quality of each of the individuals, that is, the correctness of the solution that

is represented by a given individual. As we explained before, the model has an energy associated that represents how good or bad is the adjustment with respect to the objects in the scene. Thus, the fitness function is the inverse of this energy:

$$F = 1/E(v) \tag{3.2}$$

where $E(v)$ is the energy of the mesh described in Equations 2.1 and 2.9.

3.2.1 Evolutionary process

After the definition of the two main terms of an evolutionary method, that is, the genotypic encoding and the fitness function, we can proceed with the definition of the evolutionary segmentation process.

The main steps in a standard GA evolutionary process are depicted in Figure 3.1, and basically consist of the following:

Initial population The first step is the production of the initial chromosomes. These chromosomes are randomly initialized in the search space, conforming the initial population of the evolutionary process. In our application, it means that each individual would be placed in a random part of the image, having all the nodes equidistant among them, as shown for instance in Figure 3.2.

Fitness assignment The quality of the individuals is evaluated using the fitness function. This fitness represents the quality of the possible solution that represents the individual. As we said, the inverse of the energy is used as the fitness function.

Termination condition It is checked to determine the end of the evolutionary process. In particular, if we reach a desirable segmentation or the maximum number of generations is reached, the process finishes.

Selection of individuals for reproduction Two individuals from the population are selected from the entire population, normally depending on their fitness. There are plenty of possibilities for the selection. We use *tournament window* selection, that selects a random set of individuals and picks the best individual within this set.

Crossover and genetic operators We produce the offspring by crossing the two parent chromosomes and applying other genetic operators such as mutation

and other *ad hoc* operators. The offspring are introduced in the new generation of the population. Both steps of selecting and producing new individuals are repeated, until the population is filled (using a constant size of the population). Finally, we go to the *fitness assignment* step, starting again the procedure.

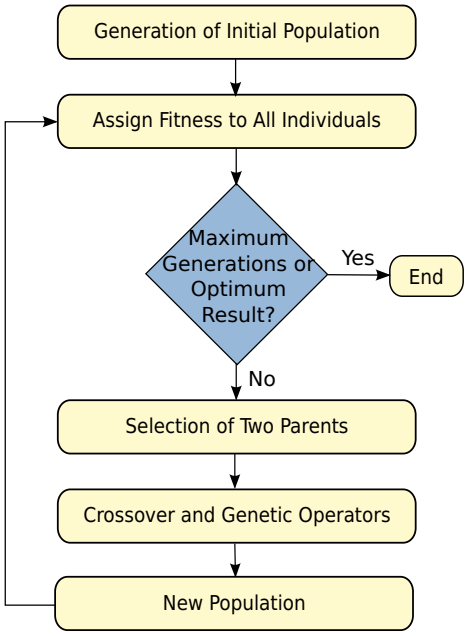


Figure 3.1: Standard genetic algorithm process.

Other important issue in an evolutionary algorithm is the *population size*. It has to be high enough to preserve the genetic diversity, that is, to cover properly the search space, but it has also not to be too high in terms of computational requirements. *Elitism* is also introduced in the method. It means that the best individual of each generation is directly copied to the next generation. With elitism, we guarantee the preservation of the best temporal result over the generations.

3.2.2 Genetic operators

One of the main aspects of an evolutionary method is the definition of the genetic operators that produce the new individuals. As we explained, we apply a set of operators to produce new individuals for the next generation. These operators have to be as much useful as possible because they determine the way the population evolves towards the optimal solution.

The classic operators in a GA, crossover and mutation, were adapted to our specific domain, but also new *ad hoc* ones were designed according to the characteristics of the model to help the convergence towards the optimum. The genetic operator examples are shown in the 3D domain, being equivalent to the 2D case.

Crossover

One of the classic operators is crossover. It consists of interchanging the information between a pair of parent chromosomes that were previously selected. The same parent chromosome can be selected several times to produce offspring, but guaranteeing that this situation does not happen too often, because, in that case the entire population can converge too fast over the same individuals, that is, focusing too fast in the same search area. In that case we loose genetic diversity, taking the risk of falling in local minima. The "selective pressure" should not be high, which can be controlled by the size of the tournament. Once we have selected the two parents, we apply the crossover with a given probability. Otherwise, the parents are preserved directly.

The *scheme theorem* is based on the idea of good construction blocks [42, 43]. A good solution of the problem is composed by good *blocks*. The crossover operator is the responsible to merge good blocks coming from different parents. The selective pressure has the relevance in order to pick good individuals from the population, but taking care in preserving the diversity in the population. The crossover has the potential to produce new good individuals after taking appropriate parents from the population.

There are several ways of merging the genotype of two parents, the most common are:

Crossover by a point The simplest way of merging two parents is selecting one random middle point of the chromosome, dividing it in two groups, and interchanging both parts between the parents to produce the genes of the new individuals.

Crossover by n points or uniform Following the same idea as the previous one, but dividing the chromosome in $n + 1$ groups and also interchanging these subgroups of genes. A particular crossover of this one is the *uniform crossover*. It consists of doing the same but at the level of the gene. Each gene of the resultant individuals is randomly selected from one of the parents.

Arithmetical crossover Other possibility consists of producing the new genes as

the weighted mean of the corresponding genes of the parents. For instance, considering x_1 and x_2 the two parents, a new gene x_i is calculated as:

$$x_i = \alpha \times x_{1_i} + (1 - \alpha) \times x_{2_i}$$

where α is a random value between 0 and 1, and weights the relevance of the value coming from each parent. This value is calculated for each gene i in the chromosome.

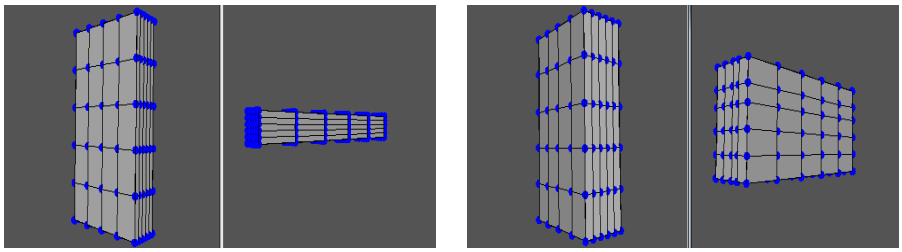
The development of the crossover operation in the used model is more difficult due to the mesh features. One of the main characteristics of the model we have to preserve consists of avoiding possible crossings in the links of the mesh or the same location for several nodes. For that reason, we had to discard the classical crossover by a point or by n points. With these operators, we could have so many individuals that do not preserve these properties. Even in this case, the evolutionary process would progressively avoid these solutions, because these individuals normally have higher energies, but having a part of the population in that situation would complicate or slow down the process.

On the contrary, the arithmetical crossover guarantees that property. If two individuals are correct (no crossings), the weighted mean of the coordinates of all the nodes will be correct. For that reason, this is the crossover we used in the evolutionary processes. As we said, an individual is the list of the coordinates of all the nodes. Thus, the coordinates of each node of the offspring is:

$$Child_1 = \begin{cases} x = \{\alpha \times x_{parent_1} + (1 - \alpha) \times x_{parent_2} \\ y = \{\alpha \times y_{parent_1} + (1 - \alpha) \times y_{parent_2} \\ z = \{\alpha \times z_{parent_1} + (1 - \alpha) \times z_{parent_2} \end{cases} \tag{3.3}$$

$$Child_2 = \begin{cases} x = \{\alpha \times x_{parent_2} + (1 - \alpha) \times x_{parent_1} \\ y = \{\alpha \times y_{parent_2} + (1 - \alpha) \times y_{parent_1} \\ z = \{\alpha \times z_{parent_2} + (1 - \alpha) \times z_{parent_1} \end{cases} \tag{3.4}$$

where z is the third dimension and it is only applied in the case of 3D segmentations. This crossover operator is not ideal, because the child does not contains exactly the genes of the parents, but the new coordinates are inspired on them, carrying that way the information about the correctness of the parents. However, this operator will preserve the integrity of the mesh and also contributes to the diversity in the population, thanks to the generation of a couple of new individuals which are "intermediate" of both parents. An example of arithmetical crossover can be seen in Figure 3.2.



(a) (b)

Figure 3.2: Arithmetical crossover operator. (a) Selected parents. (b) Offspring after the crossover.

Mutation

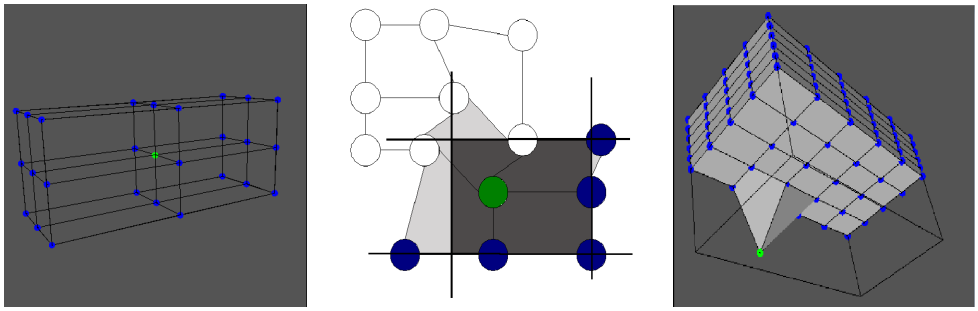
The other basic genetic operator in an evolutionary process is the mutation operator. Its main purpose is the exploration of the search space. The main idea consists of randomly change the value of a gene, that is, in our case, mutate the position of a given node. Thus, one node can reach new good positions in the search space that were not reached before.

With this operator we have the same problem as in the case of the crossover operator. Basically, we have to guarantee the integrity of the mesh. Therefore, we cannot mutate the position of the node to other random position in the entire image. Instead of this, we have to restrict the possibilities to a given subspace that guarantees the mentioned properties of the mesh.

We developed a mutation operator that avoids crossings in the model (Figure 3.3). The operator computes the limits of the node mutation, taking into account its 8 neighbor nodes in 2D and the 26 in 3D. The movements of the nodes are restrictive in order to avoid crossings. For example, in the 3D domain, the bottom limit is set by the coordinates of the northernmost neighbor of the 9 lower neighbors, and so on for each limit. Once the limits are computed, the node's coordinates are mutated at random positions within them. In the case of external nodes, virtual nodes are defined at mirrored distances of the opposite nodes in the same axis. This restrictive way of calculate the possible subspace where a node can mutate does not represents all the possible positions but represents most of them and also is faster than other possibilities to calculate the entire subspace to mutate.

Ad hoc operators

As well as the classical or standard genetic operators, that were adapted to our specific domain, other specific operators were developed, designed according to the



(a) (b) (c)

Figure 3.3: Mutation operator. (a) Representation of the 26-neighborhood of a node. (b) Delimitation of the mutation space in a 2D frame represented in dark grey. (c) Mutation of a node in an initial individual with a cubic topology.

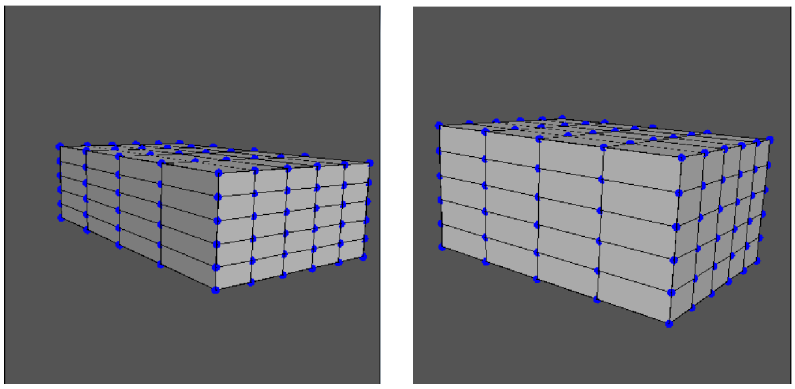
characteristics of the mesh. They were developed to complement the classic operators and to obtain better modifications in the generation of new individuals.

Spread operator Operator that in some specific moments can have its relevance. The arithmetical crossover operator used generates new individuals of the same or intermediate sizes of the parents. We commented the advantages of the operator in our application. Nevertheless, this implies that the evolutionary process is restricting the search space over the generations, having progressively smaller individuals. For that reason, we need an operator that sometimes takes an individual and produces other bigger.

The mechanism of the operator is simple. It stretches the mesh in a given direction and, therefore, the new mesh increases its size. Figure 3.4 shows an example of the application of the spread operator.

Group mutation operator Sometimes there is a mutation of a node to a good position in the image, but in terms of the global energy of the mesh it is not worth, which complicates its survival over the generations. That happens, for instance, when the mesh is quite far from an edge of the object. In this situation, one external node, with a simple mutation, is able to reach this edge. The problem is that, sometimes, the reduction of the external energy in that node, thanks to the right location of the node, is penalized by the increase of the internal energy, because the node is now farther from its neighbors.

Following this idea, we created the group mutation operator. The basic idea consists of randomly selecting a group of neighboring nodes which are mutated



(a) (b)

Figure 3.4: Spread operator. (a) Original mesh. (b) Individual generated after spreading the mesh.

simultaneously in the same direction and with the same value. Performing a group mutation is generally more useful thanks to the better placement of a group of nodes instead of a single one. Figure 3.5 shows an example of the application of the operator.

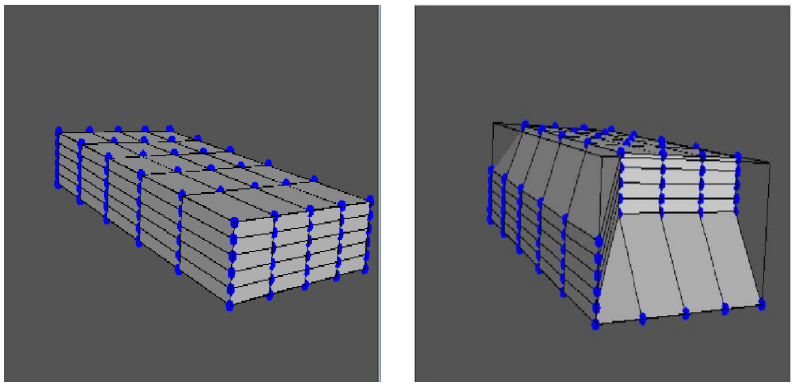
Shift operator Operator that is useful in the exploration of the search space. It moves the mesh to another position in the image. This movement allows that the external and the internal nodes can get into the object to segment at the same time approximately. This way, the position of the objects in the image does not affect the final node distribution.

This operator has its relevance especially in the first generations, where the individuals are trying to locate or cover the objects to be segmented. In Figure 3.6 it is shown a couple of individuals, the original one and the same after being moved by the shift operator.

3.2.3 Segmentation phases

The objective of the evolutionary process is to progressively obtain a set of individuals with the nodes better adjusted to the object. The internal energy terms try to produce individuals with a high level of smoothness and contraction. Thus, the segmentation of complex objects can be complicated because we take the risk that the individuals focus on parts of the object, falling the entire segmentation in local minima.

For that reason, we defined different stages or phases in the evolutionary process



(a) (b)

Figure 3.5: Group mutation operator. (a) Original mesh. (b) Individual generated after the group mutation.

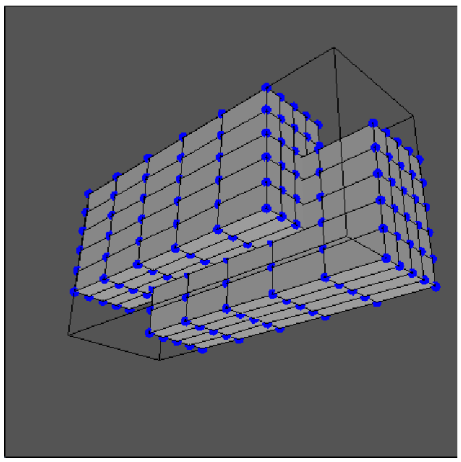


Figure 3.6: Shift operator. Original and shifted mesh after the application of the operator.

Algorithm 3.2.1: GA MINIMIZATION()

$g \leftarrow 0$ // Generation number
$P_g \leftarrow$ INITIALIZEPOPULATION()
SETPARAMETERS($explorationParameters$)
$fitness_g \leftarrow$ EVALUATEPOPULATION(P_g)
while $g < maxGenerations1stPhase$ // First evolutionary phase
do
$P_{g+1} \leftarrow$ APPLYGENETICOPERATORS(P_g)
$P_{g+1} \leftarrow P_{g+1} +$ ELITISM($P_g, fitness_g$)
$fitness_{g+1} \leftarrow$ EVALUATEPOPULATION(P_{g+1})
$g \leftarrow g + 1$
SETPARAMETERS($adjustmentParameters$)
$fitness_g \leftarrow$ EVALUATEPOPULATION(P_g)
while $error < targetError$ // Second evolutionary phase
do
$P_{g+1} \leftarrow$ APPLYGENETICOPERATORS(P_g)
$P_{g+1} \leftarrow P_{g+1} +$ ELITISM($P_g, fitness_g$)
$fitness_{g+1} \leftarrow$ EVALUATEPOPULATION(P_{g+1})
$g \leftarrow g + 1$
SETPARAMETERS($refinementParameters$)
$fitness_g \leftarrow$ EVALUATEPOPULATION(P_g)
while $g < maxGenerations3rdPhase$ // Third evolutionary phase
do
$P_{g+1} \leftarrow$ APPLYGENETICOPERATORS(P_g)
$P_{g+1} \leftarrow P_{g+1} +$ ELITISM($P_g, fitness_g$)
$fitness_{g+1} \leftarrow$ EVALUATEPOPULATION(P_{g+1})
$g \leftarrow g + 1$

with different purposes. Algorithm 3.2.1 summarizes the steps in the GA minimization process, including all this phases. For each one of them the parameter set were modified to focus on those parameters that reproduce the desirable behaviour. We defined three phases in the evolutionary process in order to cover the object or objects in the first generations and then to improve the adjustment progressively.

First phase It is an exploration phase. Its objective is to localize where the object is, producing an entire population of individuals surrounding it.

The energy parameters allow the nodes to be outside the object to segment, so individuals initialized inside the object are penalized. Thus, these individuals cannot survive in next generations. To do that, we mainly used the energy

terms In-Out (IO), that makes the internal nodes to be in the object and the external ones outside, and Gradient Distance (DG), that potentiate the individuals with the external nodes inside the object boundary. The shift operator is adequate in this aim, so it is only applied in this evolutionary phase.

Second phase It is an adjustment phase. Given a population of individuals with the external nodes surrounding the contour of the object, the purpose of this phase is to progressively obtain a correct segmentation. The parameter values are changed in order to search for a more homogeneous distribution of the nodes, reduce the size of the model, and adapt it to the image. This way, exploration predominates in the first phase while exploitation dominates the second one. The energy parameters in this phase vary depending on each segmentation process. The probability of application of the spread operator is decreased because it is not as necessary as in the first evolutionary phase. This phase finishes when the best individual cannot improve the fitness, or a given number of generations is reached.

Third phase It is a refinement phase. This phase is less relevant and optional. In this phase, the parameters of the internal energy terms (Eqs. 2.2 and 2.10) increase their values to obtain homogeneous distributions of the nodes. Also, the mutation probability of the external nodes is increased with a factor directly proportional to the gradient distance of the external nodes. This behavior is especially important in noise images, letting the external nodes improve the adjustment to the image boundaries.

In Figure 3.7, the results at the end of each phase are shown, for a specific 3D segmentation.

Regarding topological changes, it is complicated to define the possibility of topological changes in the individuals. The idea would be to insert in the genotype some information of the presence of each of the connections. However, the complexity of the methodology would be highly increased, having to deal with a large population that would cover different possible topologies and individuals to explore each subspace represented by a given topology. For that reason, the inclusion of topological changes is not allowed in this evolutionary approach.

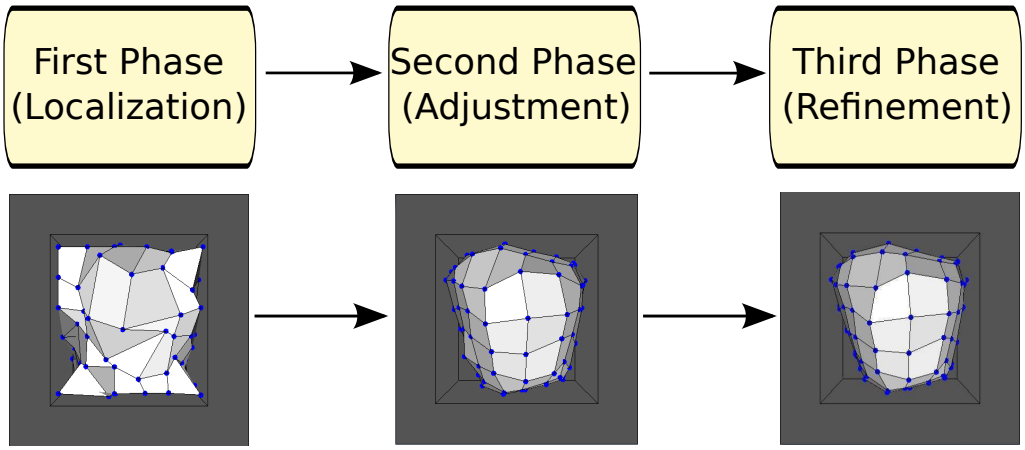


Figure 3.7: Results at the end of each phase of the segmentation process.

3.3 Results and comparison between the Genetic and Greedy Algorithms

This section presents some representative segmentation examples of several images to show the capabilities and main difficulties of the different optimization methods. Especially, most of the examples were designed to contrast the advantages of the genetic global search method that were developed against the greedy local search previously defined. Images with different characteristics and complications were used. In particular we designed some artificial images that included some difficulties like fuzzy contours, uniform noise and local noise. Thus, we want to demonstrate that the greedy local search falls in local minima meanwhile the genetic global search is capable to overcome the complications and reach an acceptable result.

In all the examples, the same image was used as the external energy for both internal and external nodes, and all the test images had 256 gray levels. As the capabilities of the different optimization algorithms are equivalent in the 2D and 3D domains, we concentrate here the analysis using examples from the 3D domain.

As we mentioned, the TAV parameters in the evolutionary algorithm are different in each phase. Table 3.1 depicts the parameters used in the first phase of the evolutionary processes, that were always the same in the different examples. These values were tuned to obtain the aim of this phase in most of the images: covering the objects to detect (using ω and ρ), with emphasis in the GD objective (high value of δ).

For the second phase of the evolutionary processes, Table 3.2 includes the TAV parameters used in the segmentation examples. The TAV parameters were exper-

Table 3.1: TAV parameter sets of the 1st evolutionary phase in the segmentation processes of the examples.

α	β	γ	ω	ρ	ξ	δ
0.00001	0.00001	0.0	1.0	1.0	0.0	10.0

Table 3.2: TAV parameter sets of the 2nd evolutionary phase in the segmentation processes of the examples.

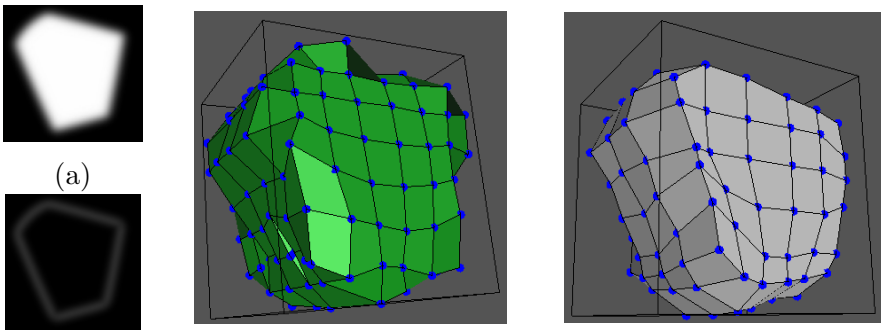
Figure	*Size*	α	β	γ	ω	ρ	ξ	δ
3.8	$6 \times 6 \times 5$	2.5	0.5	1.0	10.0	4.5	4.0	4.0
3.9	$6 \times 6 \times 5$	2.0	0.5	0.5	10.0	4.5	4.0	8.0
3.10	$6 \times 6 \times 5$	2.5	0.5	0.5	10.0	4.5	4.0	9.0
3.13	$8 \times 8 \times 7$	8.5	0.5	0.8	10.0	6.5	4.0	10.0
3.14	$8 \times 8 \times 7$	8.5	0.5	0.8	10.0	6.5	4.0	10.0
3.15	$8 \times 8 \times 7$	8.5	0.5	0.8	10.0	6.5	4.0	10.0
3.16	$8 \times 8 \times 8$	8.0	0.5	0.5	10.0	2.0	3.0	3.0
3.17	$8 \times 8 \times 8$	9.0	0.1	0.1	10.0	2.5	3.0	4.0

imentally adjusted, although the GA is less sensitive to changes in the parameter set than the greedy approach. Table 3.3 shows the GA main parameters used in the GA segmentations. We used a tournament selection with a size of 3% of the population and elitism of the best individual. The probabilities of the operators were experimentally set, too, taking values in the range where the best test results were obtained. Finally, the number of generations of the first evolutionary phase was around 100. The second phase is finished when there is not improvement in the best individual (around 1200 generations in the examples) and the third evolutionary phase includes few generations, about 100, and it is not applied in the hybrid case.

The execution time of the GA, with around 1600 individuals and a $8 \times 8 \times 8$ TAV, was usually between 5 and 6 hours in an Intel Core 2 2.4 GHz. Nevertheless, the process can be faster maintaining acceptable results if the number of generations is reduced. The processing time of the GA process in one generation depends only on the size of the net and the population, whereas the image size is not relevant. Moreover, this could be easily parallelized in several cores to speed up the process, sharing the population and the production of individuals in each generation.

Table 3.3: GA parameters used in the evolutionary processes.

Population	*p.crossover*	*p.mutation*	*p.spread*	*p.shift*	*p.group mutation*
1600	0.5	0.0005	0.01	0.05	0.001



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

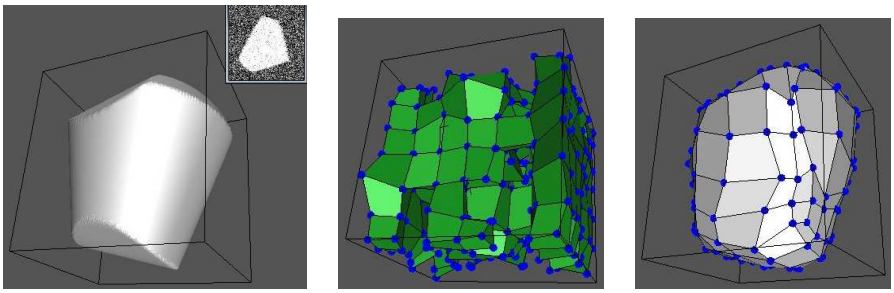
Figure 3.8: Results obtained in an image with fuzzy contours. (a) A 2D slice of the image. (b) The gradient image of the slice showed in (a). (c) Segmentation with the greedy algorithm. (d) Result with the GA.

3.3.1 Segmentation of images with fuzzy contours

The genetic algorithm can optimize the segmentation in images with fuzzy contours. Figure 3.8 shows an example with fuzzy contours. Figure 3.8 (a) represents a 2D slice of the image in the same point of view of the rest of the images. Note that the input to the algorithm, in the 3D case, is a set of consecutive 2D slices that define the 3D object, that are used to calculate the different energy terms. The number of CT slices varies from 90 to 110 in the examples, being 90 in this Figure. Figure 3.8 (b) corresponds to the gradient image of the previous slice. Note the gradual transition of external contours. The greedy algorithm does not achieve a fine adjustment to the object edges, as Figure 3.8 (c) shows, and is not able to perform a segmentation as smooth as the one obtained by the GA (Figure 3.8 (d)).

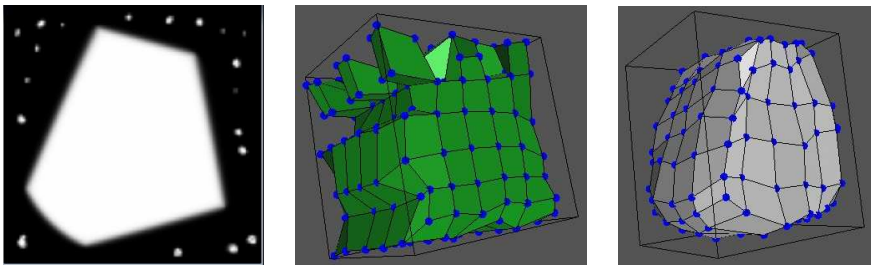
3.3.2 Segmentation of noisy images

One of the main drawbacks of the greedy search is the fall in local minima in images with noise. On the contrary, the GA, thanks to the global search, is less sensitive to the local minima problem as Figures 3.9 and 3.10 show. In the first case (Figure 3.9) a medium level of Gaussian noise was added to the original image. The greedy



(a) (b) (c)

Figure 3.9: Results obtained in an image with Gaussian noise. (a) Image to segment. The inset represents a 2D section of the image with the added Gaussian noise. (b) Segmentation with the greedy approach. (c) Segmentation with the genetic algorithm.



(a) (b) (c)

Figure 3.10: Results obtained with irregular noise. (a) A 2D slice of the original image with noise. (b) Segmentation with the greedy approach. (c) Segmentation with the genetic algorithm.

method is very sensitive to noise since it can prevent the mesh contraction so the final shape would not adequately reflect the desired segmentation. The genetic algorithm explores globally the search space so it can avoid the local minima due to noise and it can provide a correct final segmentation with a smooth node distribution of the external nodes.

In the second example (Figure 3.10), we added different spots in the background of the image of Figure 3.8, in order to test the capabilities of the GA to overcome this kind of noise. Again, the greedy algorithm stops the mesh minimization in local minima whereas the GA avoids the local minima due to noise and obtains a correct and homogeneous distribution of the external nodes of the final TAV.

3.4 Hybridization of the genetic algorithm with the local search procedures

One of the main drawbacks of the GA is the number of generations needed to converge to an acceptable result. As we said, the method needs some generations to localize the object in the scene and also needs many others to progressively adjust all the nodes to correct positions. Moreover, sometimes there are some objects that present complex surfaces, whereas other images present more than one object. With the original topology, the model has many restrictions to provide a correct adjustment to all the characteristics of the segmentations required. For that reasons, we need to introduce in the process some mechanism, firstly, to fasten the process and, secondly, to allow the mesh more flexibility to perform better adjustments in complex segmentations.

We combined the GA global search and the greedy local search (explained in Chapter 2), by means of a *Lamarckian* strategy. This is, the greedy search is applied to each individual of the genetic population, typically a short number of greedy steps. A step is defined as the application of the greedy minimization to all the nodes of the mesh. As a result, the fitness of the individuals changes. A combination using a Lamarckian strategy means that the changes in the structures provided by the greedy search revert to the original genotypes. Thus, all the advances performed by the greedy method are kept in the population so the convergence is faster. Algorithm 3.4.1 describes the main process of the hybrid combination.

Other advantage of the hybrid approach is that it overcomes the limitation of the GA implemented related to its inability to perform topological changes in the mesh, provided by the greedy method. The combined model uses the procedures of the local search to cut links between adjacent external nodes after the minimization process. Figure 3.11 shows an example where the link cutting procedure is applied to an individual of the genetic population. The object in the image contains a hole that crosses the entire object from one side to the other. As it is shown in the images, the method progressively cuts connections until a correct detection of the hole is obtained.

As we explained in Chapter 2, the link cutting mechanism performs the cuts with coherence to obtain an homogeneous process of breaking a group of links. This is useful in images like Figure 3.11, to be able to detect the entire hole, or also to perform the division of the mesh if we have more than one object. In this case, the method has to be able to divide the mesh and perform the segmentation for each one of them. Figure 3.12 shows other examples where the net division procedure

Algorithm 3.4.1: GA-greedy algorithm hybridization(*Population*)

```
for each Individual ∈ Population
do { steps ← GenerateRandom()// between 0 and max number of steps
     for i ← 0 to steps
     do { updatedIndividual ← false
          for each node ∈ Model  // Greedy step
          do { newPosition ← false
               E_local ← ComputeLocalEnergy(node)
               neighborPixels ← GetNeighborPixels(node)
               for each pixel ∈ neighborPixels
               do { newE_local ← ComputeLocalEnergy(pixel)
                    if newE_local < E_local
                    then { E_local ← newE_local
                           newPixel ← pixel
                           newPosition ← true }
               if newPosition
               then { SetCoordinates(node, newPixel)
                      updatedIndividual ← true }
          if not updatedIndividual
          then TopologicalChanges() }
```

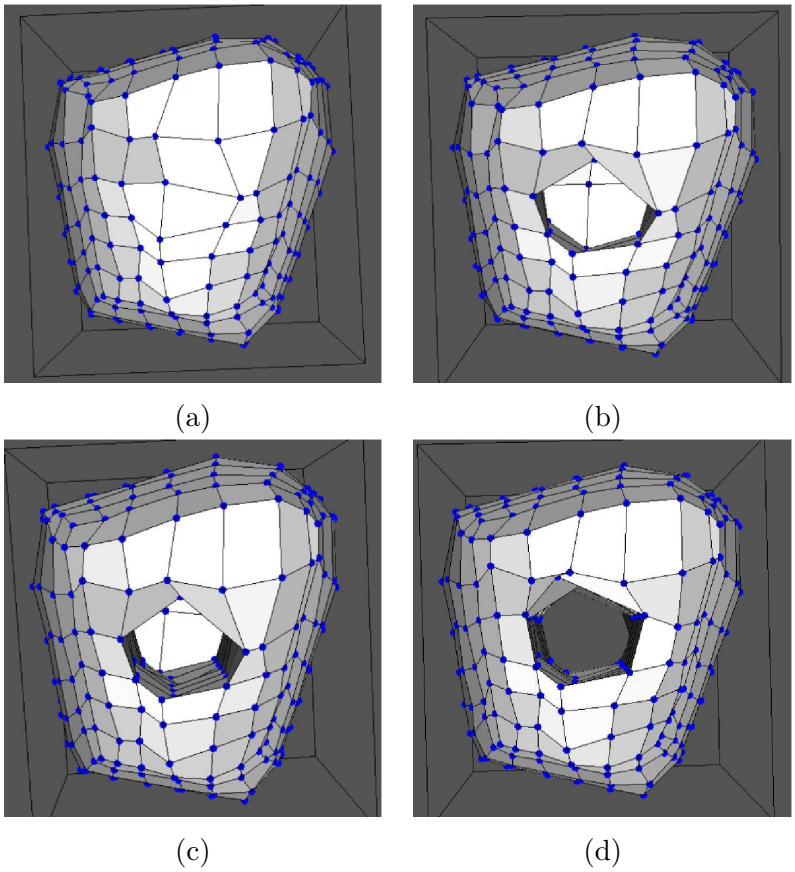
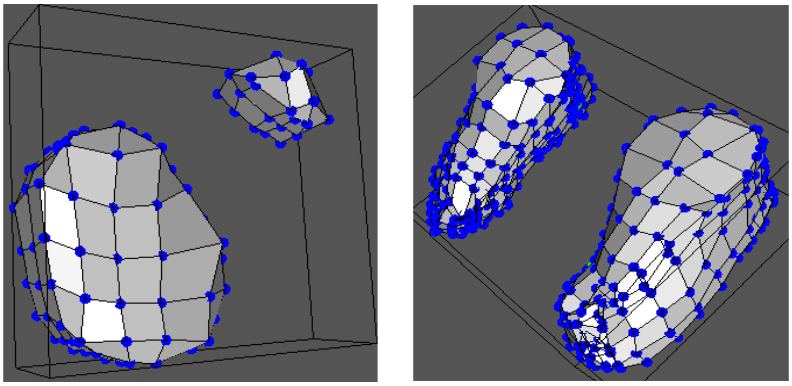


Figure 3.11: Example of segmentation with a breaking sequence. (a) Individual before breaking. (b) and (c) Intermediate steps. (d) Final result after breaking.



(a) (b)

Figure 3.12: Example of segmentation with several objects in the scene. (a) Segmentation of a synthetic image with two objects.(b) Segmentation of a CT image with two feet.

is needed, so it acts over the individuals of the population. In both examples, corresponding to a synthetic and a real CT image, the hybrid method was capable to divide the mesh and perform the simultaneous segmentation of both objects in the scene.

Throughout the generations, the topology of the best individual is considered in the rest of the population of the next generation. It means that, if the topology of an individual changes by the application of the link cutting or automatic division procedures, and it is the best individual, then such topology is "exported" to the other individuals. Additionally, in our implementation, the iterations of the greedy search (with the associated link cutting and net division procedures) are only performed in particular generations of the evolutionary process, beginning in the second evolutionary phase. That number was typically a random value between 1 and 6. The reason is to incorporate all the advantages of the local search procedures avoiding a high level of exploitation provided by such methods.

3.5 Results with the hybrid approach

The hybrid method was also tested using examples that require topological changes through the link cutting and automatic net division procedures. We tested the advantages of the hybrid algorithm that combines the GA with these procedures.

The combination with the local procedures occurs only in the second evolutionary phase. Moreover, as we explained, these procedures are not applied in all generations. We used them every 25 generations. When the procedures were applied

to each individual of the genetic population, a small number of steps was used, a random number between 1 and 6. Remember that a greedy step is defined as the application of the greedy movements to all the nodes of the mesh. Therefore, we did not use a great level of exploitation with the local search, which can contribute to fall in local minima.

We also compared the hybrid approach with the greedy local search, following the same idea as used with the GA approach. Once again, we designed some specific images that present difficulties for the segmentation process, that is, fuzzy contours, uniform noise and local noise. In this case we introduce a hole in the objects, requiring topological changes. This experiments were designed to demonstrate if the different approaches are capable to overcome the different complications in the image and, at the same time, if the hole is correctly detected. Finally, we tested the methods with some CT images taken from the medical domain that require all the described characteristics. Thus, the images has different levels of noise and also require topological changes or the division of the mesh to detect different objects.

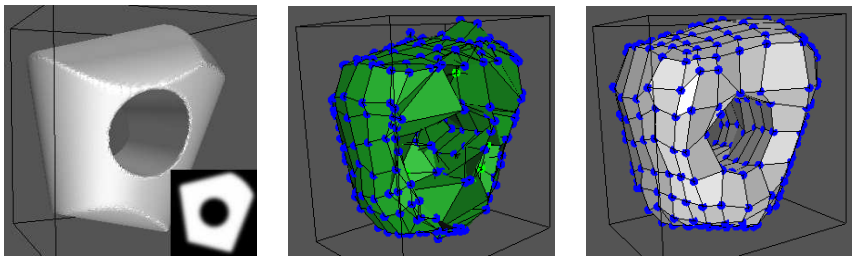
The experiments were done under the same characteristics as depicted in Section 3.3. Regarding the energy terms, Tables 3.1 and 3.2 also include the TAV parameters used in these segmentation examples whereas Table 3.3 shows the GA main parameters used in the GA segmentations.

3.5.1 Segmentation of images with fuzzy contours which require topological changes

The first example is an image with an inner hole (Figure 3.13). Additionally, its external contour is fuzzy as the slice of the original image inset shows. The greedy search begins the link cutting but it gets stuck in local minima and does not obtain an accurate final segmentation due to the fuzzy contours (Figure 3.13 (b)). The hybrid approach combines the advantages of both global and local strategies, since the global search avoids the fuzzy or noisy contours and the local search allows the link cutting procedure. This way, the final TAV delimits perfectly the hole as Figure 3.13 (c) shows.

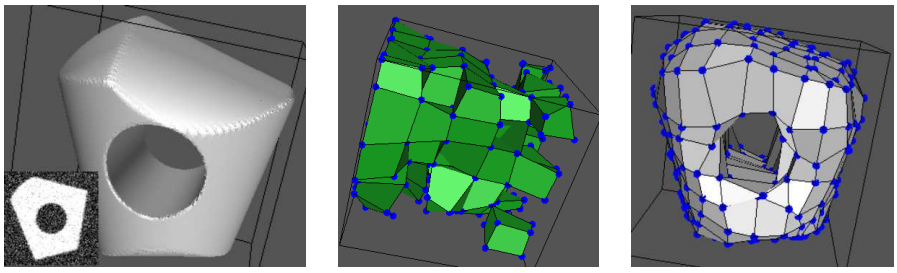
3.5.2 Segmentation of images with noise which require topological changes

We also tested the hybrid method in images with different kinds of noise. Figure 3.14 shows the results obtained with the same image as in the previous subsection. In this case, a medium level of Gaussian noise was added to the original image. It is clear that the greedy method is very sensitive to noise, whereas the hybrid method



(a) (b) (c)

Figure 3.13: (a) Image with fuzzy external contours. (b) Segmentation with the greedy approach. (c) Segmentation with the hybrid algorithm.

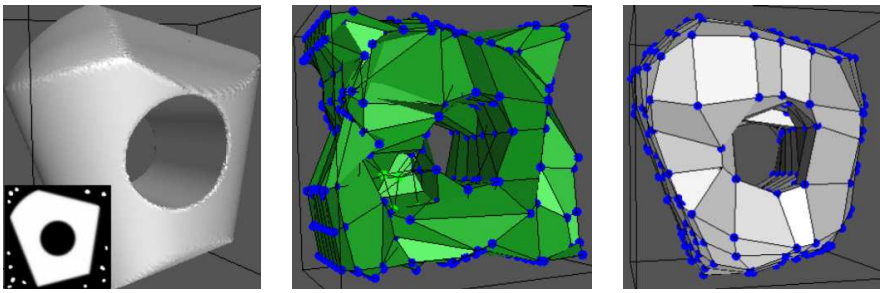


(a) (b) (c)

Figure 3.14: (a) Results obtained with an image with Gaussian noise. (b) Segmentation with the greedy approach. (c) Segmentation with the hybrid algorithm.

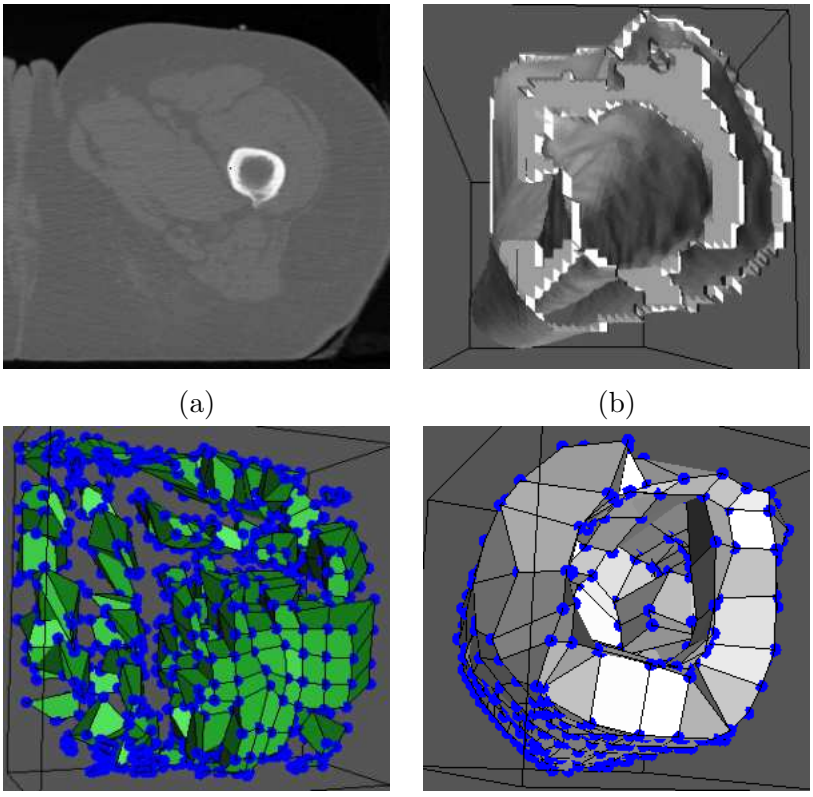
overcomes the noise problem. The example in Figure 3.15 includes several spots in the original image. Once again, the results indicate that the greedy method cannot achieve the desired segmentation. Nevertheless, the hybrid combination overcomes this type of noise at the same time that performs correctly the topology adaptation through the link cutting procedure.

Figure 3.16 presents an example in the medical domain. In this case, the hybrid process is tested with a humerus composed by CT slices, as the one shown in Figure 3.16 (a). Note the high level of noise of the CT slice. As the aim is the segmentation of the bone, the optimization algorithm must overcome the external contour corresponding to the flesh, in addition to the noise surrounding the bone. Figure 3.16 (b) is a 3D reconstruction from the 2D slices. Once again, the greedy algorithm is not able to achieve a fine segmentation (Figure 3.16 (c)) meanwhile the hybrid algorithm obtains a correct result in this type of real images with noise and fuzzy contours (Figure 3.16 (d)).



(a) (b) (c)

Figure 3.15: (a) Image with irregular noise. (b) Segmentation with the greedy approach. (c) Segmentation with the hybrid algorithm.



(a) (b)

(c) (d)

Figure 3.16: (a) Slice of the CT images set. (b) 3D representation of the humerus. (c) Segmentation with the greedy approach. (d) Segmentation with the hybrid algorithm.

3.5.3 Segmentation of images with several objects

One particular case in the segmentations which require topological changes is the net division in subnets to segment several objects in the image. We tested the hybrid version in artificial and real images with several objects. Figure 3.17 shows an example of segmentation in a real domain. The image is composed with a sequence of CT images that contain two bones, a tibia and a fibula. Figure 3.17(a) represents a slice of this CT image set. Figure 3.17(b) shows the 3D reconstruction from the 2D slices. In addition of the fuzziness of the contours of the two bones, the external contour of the leg introduces a contrast in the background gray level that the algorithms must overcome. Due to this, the greedy approach cannot achieve a correct segmentation (Figure 3.17(c)) meanwhile the hybrid algorithm overcomes the external contour and the image noise to provide a correct division of the subnets (Figure 3.17(d)). Note that the bigger bone requires the link cutting procedure to segment the hole of its internal part. The other bone is very small for a perfect representation and would require a mesh with a higher node density.

3.6 Discussion

We presented new approaches to the energy minimization task in the Topological Active Models. Genetic algorithms were used in conjunction with the TAV and TAN models in order to find the lowest mesh energy, this is, the best fit to the scene objects. We proposed new operators, whereas the classic ones, such as the crossover or the mutation operator, were adapted to the problem. In addition, we used a hybrid Lamarckian combination of the greedy local search with the global search of the genetic algorithm.

The genetic algorithm developed was tested with several images. The set of images used presented different difficulties to demonstrate the robustness of the global search method that was proposed, especially in comparison with the greedy local search that falls in local minima as soon as complications arise in the image to be processed. In all the examples, the new approach achieved a correct adjustment to the objects and improved the results of the greedy algorithm. The genetic algorithm was not sensitive to noise and it obtained correct segmentations in images with fuzzy contours. The hybrid approach was useful in images that require the use of the topological changes provided by the local search, together with the advantages of the global search. The approach obtained correct segmentation results in images with several objects or complex surfaces.

Table 3.4 compares the main characteristics of the proposed genetic approaches

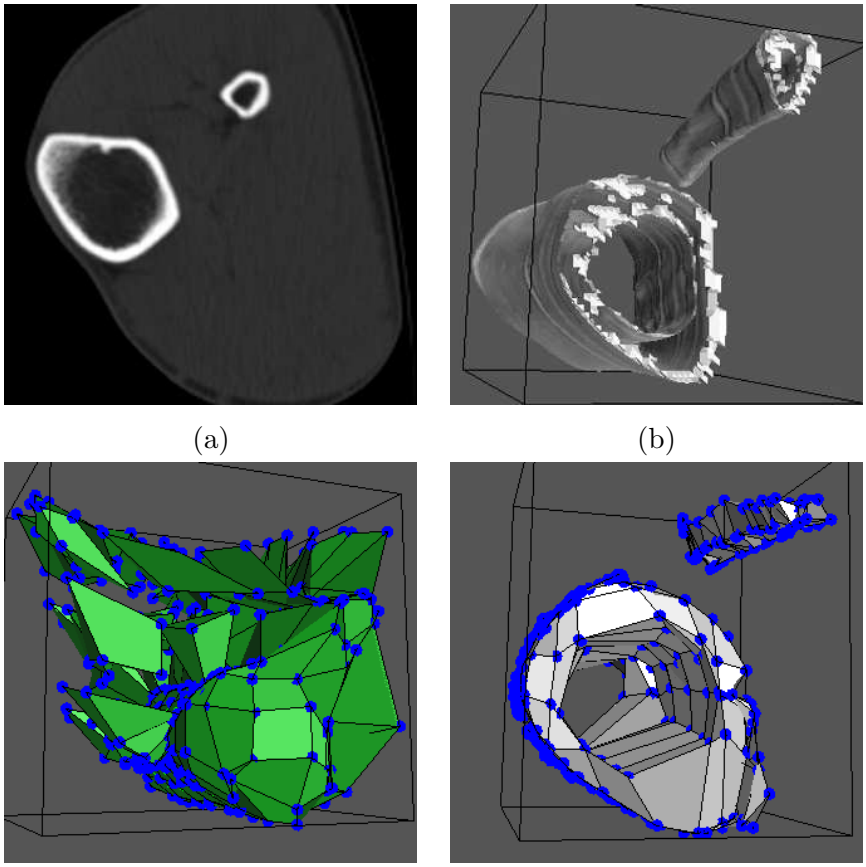


Figure 3.17: Image with several objects. (a) Slice of the CT images set. (b) 3D representation of the tibia and fibula. (c) Segmentation with the greedy approach. (d) Segmentation with the hybrid algorithm.

Table 3.4: Comparison between the greedy local search and the GA approaches.

	Greedy method	**GA method**	**Hybrid GA-Greedy**
Boundary detection	Good	Very good	Very good
Sensitivity to fuzzy boundary	Very high	Very low	Very low
Sensitivity to noise	Very high	Very low	Very low
Topological Changes	Yes	No	Yes
Execution time	Very low	High	Medium

and the greedy method previously implemented.

The main advantage of the proposed GA approaches remains in the robustness, as we could see in the segmentations in complex images. The proposed methods can reach acceptable results working with different complications in the image, like noise or fuzzy contours. However, the proposed methods presents some drawbacks. The most important one is the complexity of the method. We have to use a large set of genetic operators that evolve a population during a large number of generations until we can reach an acceptable result. Moreover, as other single-objective minimization techniques, we need to tune the energy parameters to choose those ones that provide the desirable results. The approaches presented in next chapters are introduced to solve or minimize these problems.

Chapter 4

Optimization of Topological Active Models by means of Differential Evolution

4.1 Introduction

The proposed GA provides correct results working under different kind of conditions, as we explained in Chapter 3 using images with the presence of fuzzy contours or different kinds of noise. However, the proposed method presents some specific drawbacks. One of the main problems comes from the set of genetic operators that were proposed in the method. Each of the operators has a probability associated that needs to be tuned for each possible segmentation. Moreover, it is desirable to obtain faster convergences in the segmentations with respect to the slow convergence of the GA. For that reason, we developed different improvements in the evolutionary approaches used.

One of these improvements was the use of Differential Evolution as an optimization method of the segmentation model. Differential Evolution is an alternative evolutionary method [77, 78] that minimizes the decisions of the designer with respect to others such as genetic algorithms. This new approach substitutes the entire set of genetic operators of a GA for the new one defined in Differential Evolution. The rest of the steps of application of the evolutionary algorithm to the optimization of the segmentation model are the same as the ones proposed in the basic GA already explained.

Furthermore, we hybridized Differential Evolution with the greedy search to integrate the advantages of global and local searches at the same time that the seg-

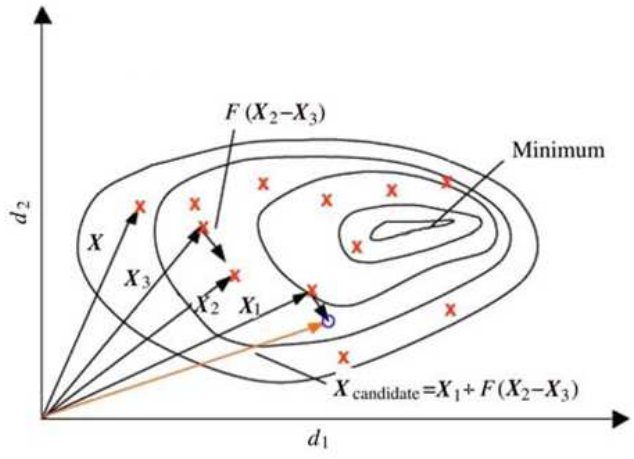


Figure 4.1: Example of 2D calculation of a candidate solution ($X_{candidate}$) to replace a given individual X.

mentation speed is improved, following the same idea as the hybridization proposed with the genetic algorithm. We also included in the local search the possibility of topological changes to perform a better adjustment in complex surfaces, topological changes that introduce the necessary mechanism to divide the mesh in the case of the presence of several objects in the scene.

4.2 Differential Evolution

Differential Evolution (DE) [77, 78] is a population-based search method. DE creates new candidate solutions by combining existing ones according to a simple formula of vector crossover and mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. The central idea of the algorithm is the use of difference vectors for generating perturbations in a population of vectors. This algorithm is especially suited for optimization problems where possible solutions are defined by a real-valued vector. The basic DE algorithm is summarized in the pseudo-code of Algorithm 4.2.1. For each individual x of the population, DE determines a candidate y, using the difference vector between two different vectors x_2 and x_3 (randomly chosen, which is summed to a base vector x_1). If the new vector is the best in terms of fitness, then it replaces the x vector. So, the population converges progressively to the best optimums found. Figure 4.1 shows a 2D example that illustrates the different vectors that are involved in the production of a given candidate $X_{candidate}$ for each individual X.

Algorithm 4.2.1: DIFFERENTIAL EVOLUTION(*Population*)

for each $Individual \in Population$
do $\{ Individual \leftarrow$ INITIALIZERANDOMPOSITIONS()
repeat
for each $Individual \;\; x \in Population$
do
- $x_1, x_2, x_3 \leftarrow$ GETRANDOMINDIVIDUAL($Population$)
- // must be distinct from each other and x
- $R \leftarrow$ GETRANDOM$(1, n)$ // the highest possible value n is the
- // dimensionality of the problem to be optimized
- **for each** $i \in 1 : n$
- // Compute individual's potentially new position $y = [y_1, ..., y_n]$
- **do**
  - $r_i \leftarrow$ GETRANDOM$(0, 1)$ // uniformly in open range (0,1)
  - **if** $((i = R) \;||\; (r_i < CR))$ $y_i = x_{1_i} + F(x_{2_i} - x_{3_i})$
  - **else** $y_i = x_i$
- **if** $(f(y) < f(x))$ $x = y$ // replace x with y in Population

until TERMINATIONCRITERION()
return (GETLOWESTFITNESS($Population$)) // candidate solution

One of the reasons why Differential Evolution is an interesting method in many optimization or search problems is the reduced number of parameters that are needed to define its implementation. The parameters are F or differential weight and CR or crossover probability. The weight factor F (usually in $[0, 2]$) is applied over the vector resulting from the difference between pairs of vectors (x_2 and x_3). CR is the probability of crossing over a given vector (individual) of the population (x) and the candidate vector y. Finally, the index R guarantees that at least one of the parameters (genes) will be changed in such generation of the candidate solution.

As Feoktistov [34] indicates, the fundamental idea of the algorithm is to adapt the step length ($F(x_2 - x_3)$) intrinsically along the evolutionary process. At the beginning of generations the step length is large, because individuals are far away from each other. As the evolution goes on, the population converges and the step length becomes smaller and smaller.

In our application each individual encodes a topological model. The genotypes code the Cartesian coordinates of the nodes. If a component of a mutant vector (candidate solution) goes off its limits, then the component is set to the bound limit. In this application it means that, in order to avoid crossings in the net structure, each node coordinate cannot overcome the limits established by its neighbors.

Moreover, the usual implementation of DE chooses the base vector x_1 randomly or as the individual with the best fitness found up to the moment (x_{best}). To avoid the high selective pressure of the latter, the usual strategy is to interchange the two possibilities across generations. Instead of this, we used a tournament to pick the vector x_1, which allows us to easily establish the selective pressure by means of the tournament size.

4.3 Differential evolution results and comparison with a genetic algorithm

This section presents some representative segmentation examples of several images. The examples were selected to show the capabilities of the proposed method and also to compare it with respect to the previous GA approach. In all the examples, the same image was used as the external energy for both internal and external nodes, and all the test images had 256 gray levels. In the 3D case, the input to the algorithms is a set of consecutive slices with such properties. In the 3D examples we used between 90 and 200 slices in the different examples. The selected 2D and 3D segmentation examples show the capabilities and advantages of the DE approach and its hybridization with a greedy local search, including noisy images, segmentations

with necessary topological changes and with the need of the division of the mesh.

All the processes used a population of 1000 individuals and took a number of generations between 400 and 2000. Regarding the DE algorithm, the tournament size to select the base individual x_1 in the DE runs was 3% of the population. We used a fixed value for the CR parameter (1.0), whereas we used a maximum value of 0.6 for the F parameter. In the different applications of the equation which determines a candidate solution (Algorithm 4.2.1), we used a random value for F between 0.2 and such maximum value (for each node), parameters that were experimentally tuned to provide the best results in most of the images. This allows that each node can move its position in a different intensity, although in the direction imposed by the difference vector $(x_2 - x_3)$, which facilitates that each node can independently fall in its best location, such as the object boundaries in the case of the external nodes. This strategy provided us with the best results in all the images.

Table 4.1 includes the energy parameters used in the segmentation examples. Those were experimentally set as the ones in which the corresponding algorithm gave the best results for each kind of image. If two evolutionary phases are used (as explained next), these parameters correspond to the second evolutionary phase.

First, we tested the advantages of DE in our application compared with a classic Genetic Algorithm (GA). In the previous chapter we defined the genetic operators used with a GA: arithmetic crossover, mutation of a node, mutation of a group of neighboring nodes, shift of a mesh and spread of a mesh. We also reasoned about the definition of two evolutionary phases that are necessary to obtain correct results in any image. The two phases are: a first one, whose aim is to produce a population of individuals that cover the object in the image, and a second one, with a different set of energy parameters, to refine the adjustment. So, the first phase provides a rough boundary detection meanwhile the second phase provides a better boundary segmentation and a better distribution of nodes. In the first phase, to obtain a rough boundary detection, the parameters were tuned giving high importance to the gradient distance energy term meanwhile the internal energy parameters took low values. These two phases are necessary as with only one phase it is easy to fall in local minima because the internal energy parameters tend to compress the mesh. This problem can be seen in Figure 4.2, $1^{st}row$, with an example in the 2D domain for a better comprehension, where the best individuals only cover a part of the object. Meanwhile, the classic GA process with the two phases, can firstly cover the object and secondly refine a final correct result, as shown in Figure 4.2, 3^{rd} row.

The problem of falling in local minima, as shown in Figure 4.2, $1^{st}row$, can be solved with the initialization of the population with individuals with a minimum

Table 4.1: TAN and TAV parameter sets used for the segmentation of the image examples (γ is only used in 3D cases).

Figure	*Size*	α	β	γ	ω	ρ	ξ	δ	τ
4.2	10×8	1	2.5	-	10	4	5	10	20
4.3	8×8	3.5	5	-	10	4	5.5	8	10
4.4,4.5	$5 \times 5 \times 5$	0.5	10	0.1	10	4.5	5.0	8	20
4.6	8×8	2	5	-	10	4	5	6	40
4.7	$5 \times 5 \times 10$	2	5.5	1	10	4	5	10	10
4.8 (a)	10×10	5	1	-	10	4	1	20	10
4.8 (b)	10×10	3	5	-	1	4	0	5	30
4.8 (c)	8×8	5	2	-	20	4	0	10	10
4.8 (d)	8×8	1.5	2.5	-	10	4	1	5	5
4.9	$12 \times 12 \times 12$	5.5	2.5	2.5	10	4.5	1	10	5
4.10 (a)	14×11	10	1	-	10	4	5	2	0
4.10 (b)	9×9	10	1	-	10	4	5	10	4
4.10 (c)	18×17	7	2	-	10	4	2	10	4
4.11 1^{st} row	$9 \times 7 \times 6$	0.1	2.5	0.1	10	2	1	20	5
4.11 2^{nd} row	$10 \times 10 \times 5$	5	4	2.5	10	4.5	0.1	10	0
4.12 (a),(b)	15×15	10	1	-	10	4	5	2	0
4.12 (c)	15×15	10	1	-	10	4	0	10	10
4.13 1^{st} row	$16 \times 6 \times 8$	0.5	1.5	0.5	10	4.5	0.1	10	2
4.13 2^{nd} row	$8 \times 8 \times 6$	10	0.0001	0.0001	10	2.5	1	5	0

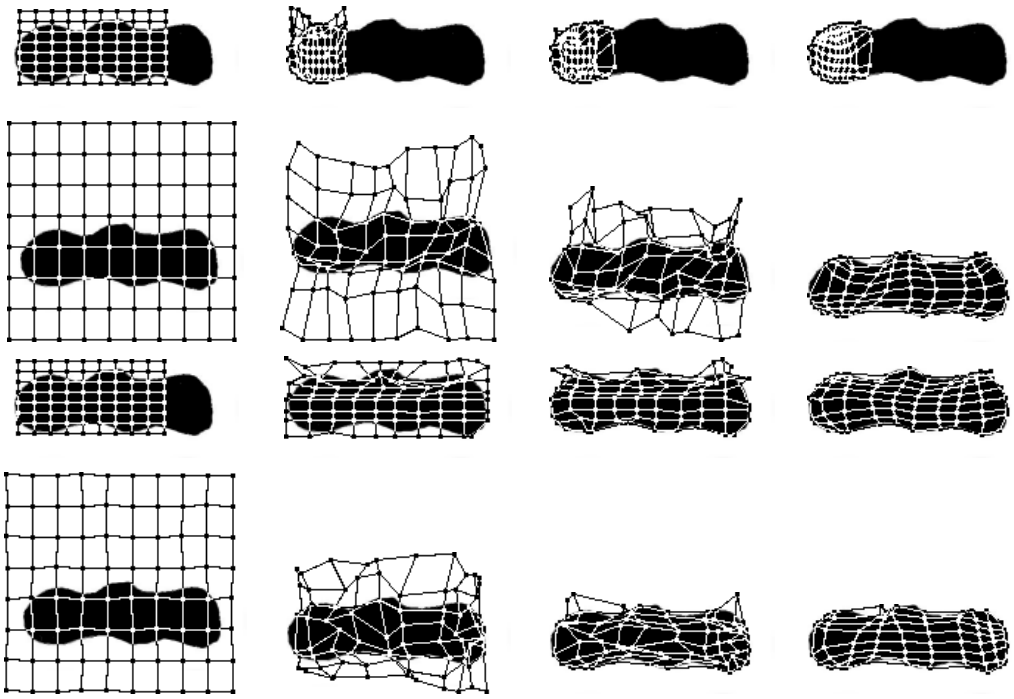


Figure 4.2: Comparison of evolutionary methodologies in 2D. The four figures in each row correspond with the best individual at different generations, from the initial generation to the best found result at the end of the evolutionary process. 1^{st} row, classic GA process, one evolutionary phase and initial random individual sizes. 2^{nd} row, classic GA process, one evolutionary phase and initial large individual sizes. $3^{rd}row$, classic GA process, two evolutionary phases. 4^{th} row, differential evolution process.

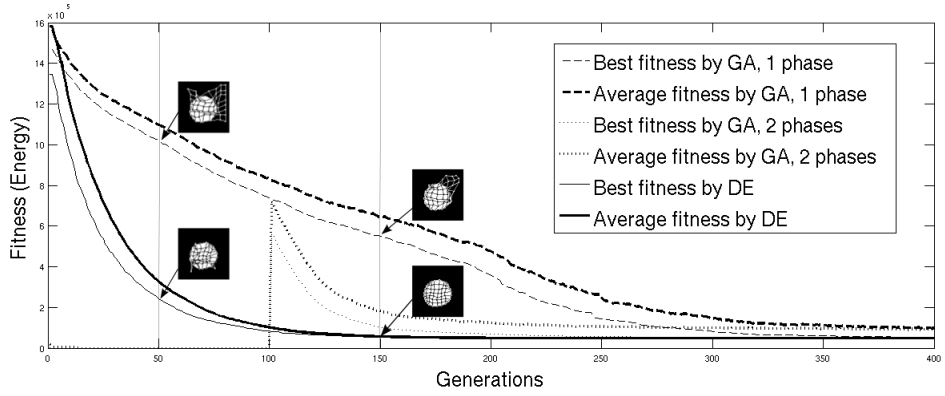


Figure 4.3: Best individual fitness (energy) and average fitness of the population with the different evolutionary processes in 2D. The curves are an average of 20 different runs with different initial populations. Insets: Best temporal result after the given number of generations.

large size. Thus, these individuals can cover the possible objects present in the scene avoiding the possible falls in parts of them. An example of this solution with the GA process with only one phase can be seen in Figure 4.2, $2^{nd}row$, where the individuals cover the entire object and, thanks to the mutation and group mutation operators, converge progressively to the object contour. However, this solution implies a very slow convergence, as we show later. Nevertheless, the same idea is useful with the proposed differential evolution process. Hence, we can tune only one set of energy parameters to be used with only one evolutionary phase. An example of evolution of the best individual with DE is shown in Figure 4.2, $4^{th}row$.

We simplified the genetic process with the proposed DE methodology because the set of genetic operators (crossover, mutation, group mutation, spread and shift of the GA) was avoided. Moreover, the convergence was faster. An example of comparison in 2D is shown in Figure 4.3. This Figure compares the evolution of the best energy (fitness) and the average energy of the population over the generations using DE, a GA with only one phase and a GA with two evolutionary phases, using the initializations explained before and using a simple artificial image. In the last case, the graph shows only the evolution of the second phase, from generation 100, as the energy of the first phase is not comparable because of the different energy parameters. Moreover, these fitness evolutions were the result of an average of 20 different evolutionary processes with different initial populations.

As it can be seen, the convergence of the GA process with one phase (dashed lines) is the worst because the large initial individuals have to be progressively

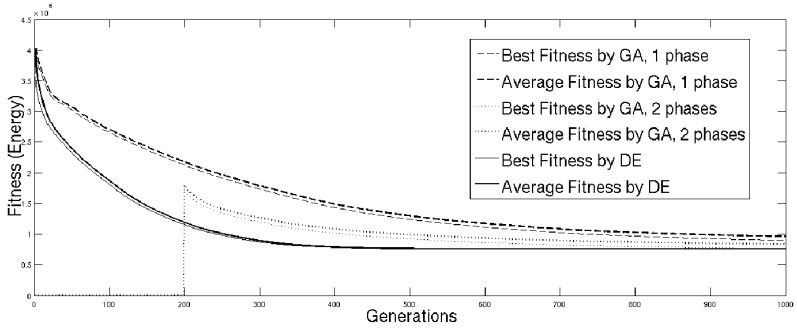


Figure 4.4: Best individual fitness and average fitness of the population with the different evolutionary processes in 3D. The curves are an average of 20 different runs with different initial populations.

approximated to the boundary of the object. As explained before, this is solved with the definition of the two different evolutionary phases in the GA process with different tasks (dotted lines). However, with the DE process, with only one phase and initial large individuals we obtain a significant faster convergence (solid lines).

We performed the comparison of the evolutionary methodologies in the 3D domain following the same idea of the 2D case. Figure 4.4 compares the evolution of the energy of the best individual and the average energy of the population over the generations, using again DE, a GA with only one evolutionary phase and a GA with two phases. As in the 2D case, we performed 20 different runs of the algorithms to obtain the average quality evolutions shown in the Figure. This comparison was made testing the different approaches in the object of Figure 4.5, using the same energy parameter set in all cases.

In Figure 4.5 the best individual at different generations with all the mentioned processes is shown. We can see intermediate results in generations 50 and 200 (1^{st} and 2^{nd} columns) and the final result (3^{rd} column). As it can be seen in the intermediate results, the GA with only one phase and large initial individuals (1^{st} row) moves slowly the external nodes to the object boundary, situation that is overcome with the GA with 2 phases (2^{nd} row). In this case, the first phase is focused on surrounding the contour of the object so we obtain the external nodes well placed in less generations. However, the DE approach, with only one phase and large initial individuals (3^{rd} row), thanks to the way that it produces new individuals can quickly obtain a population surrounding the object, producing a faster correct segmentation with a correct distribution of nodes.

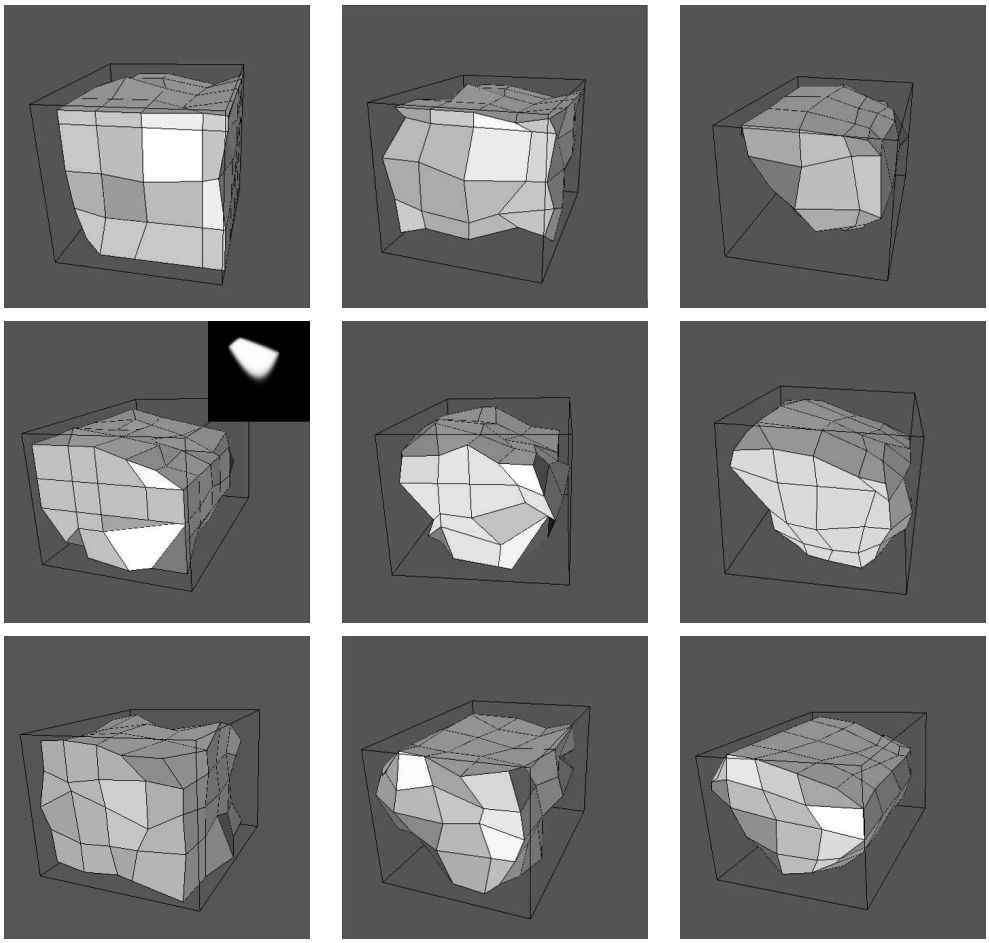


Figure 4.5: Best individual across generations (50, 200 and final generation) in different evolutionary processes in 3D. Inset in 2^{nd} row: slice of the original image. 1^{st} row, classic GA process, one evolutionary phase. 2^{nd} row, classic GA process, two evolutionary phases. $3^{rd}row$, differential evolution process.

4.4 Hybridization of the evolutionary and local search algorithms

Once we demonstrated the faster convergence of the new evolutionary approach, we proceeded to hybridize the DE method with the greedy local search, following the same methodology as the combination performed with the GA. As we explained in the previous chapter, the main idea was to integrate the advantages of global and local searches, and to obtain faster segmentations. A number of greedy steps was applied in all the genotypes of the population used by the DE process. A greedy step implies the application of the greedy movements in all the nodes of the encoded model. The number of steps was a small number (randomly between 0 and 4) to minimize the falling in local minima. Moreover, with this aim, the greedy algorithm was applied only in particular generations of the evolutionary process, typically every 10 generations.

Once again, this combination also introduces, thanks to the greedy approach, the link cutting procedure and the automatic division of a mesh. As explained before, it gives flexibility to the segmentations by breaking connections to perform better adjustments. In particular, the link cutting procedure is applied to the best individual of the population when the greedy method cannot perform more movements. The resulting topology is extrapolated to the entire population.

4.5 Hybridization of differential evolution and the greedy search results

We combined DE with the greedy local search procedures, detailed in the previous chapter, with two aims: to integrate the advantages of global and local searches, and to obtain faster segmentations. As in the hybridization with a GA, the hybrid approach followed a Lamarckian strategy since the results of the greedy search revert in the original genotypes used by DE. The characteristics and parameters of the segmentation examples are following the same idea as depicted in Section 4.3. Regarding the energy terms, Table 4.1 also include the different TAM parameters used in these segmentation examples.

Figure 4.6(a) shows different evolutions of the best individual over the generations using different configurations of the hybrid approach and compared with the DE and the greedy approach, in the 2D domain. In this case, the greedy steps were applied to the individuals each 10 generations. The graphs of fitness evolution were an average of 10 evolutionary runs of the corresponding algorithm with different

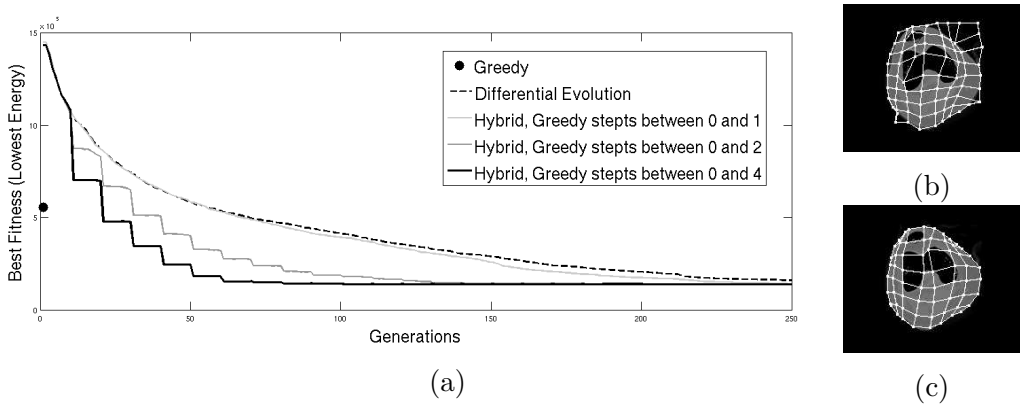


Figure 4.6: (a) Best individual evolution comparing the DE approach with different hybrid approaches and the greedy method in 2D. (b) Best final result with the greedy method. (c) Best final result of the hybrid approach with greedy steps between 0 and 4.

initial populations. As the graphic shows, the more greedy steps are used, the faster the energy minimization is, but the higher is the predominance of the greedy minimization with respect to the DE minimization. So, we can use a hybrid combination that uses a relative small number of greedy steps to speed up the process without penalizing the robustness of the DE methodology. Finally, we point out the poor segmentation provided by the greedy approach that gave a result with several nodes stuck in the local noise included in the image used (Figure 4.6(b)). This is a CT image of the knee with external noise, where the internal nodes must escape from the internal knee bones. Meanwhile, the hybrid approach (with greedy steps between 0 and 4 in this case) was able to overcome these difficulties (Figure 4.6(c)).

The same comparison was performed in the 3D case. Figure 4.7, upper graph, shows different evolutions following the same methodology as in the 2D case applying the greedy steps every 10 generations. The example corresponds to a 3D image of a humerus composed by CT slices, where the nodes have to overcome the flesh surrounding the bone. The greedy approach provided a poor segmentation with all the nodes stuck in the flesh contour (Figure 4.7(b)). Meanwhile, the hybrid approach (Figure 4.7(d), with greedy steps between 0 and 4 in this case) was able to overcome these difficulties, obtaining even a better segmentation than DE alone (Figure 4.7(c)). The greedy method, incorporated in the hybrid approach, also helps to obtain a more homogeneous internodal distribution and even to find a better adjustment through the external nodes with respect to the DE method. This can be seen in Figure 4.7, upper part, where the hybrid combinations reached a lower energy value.

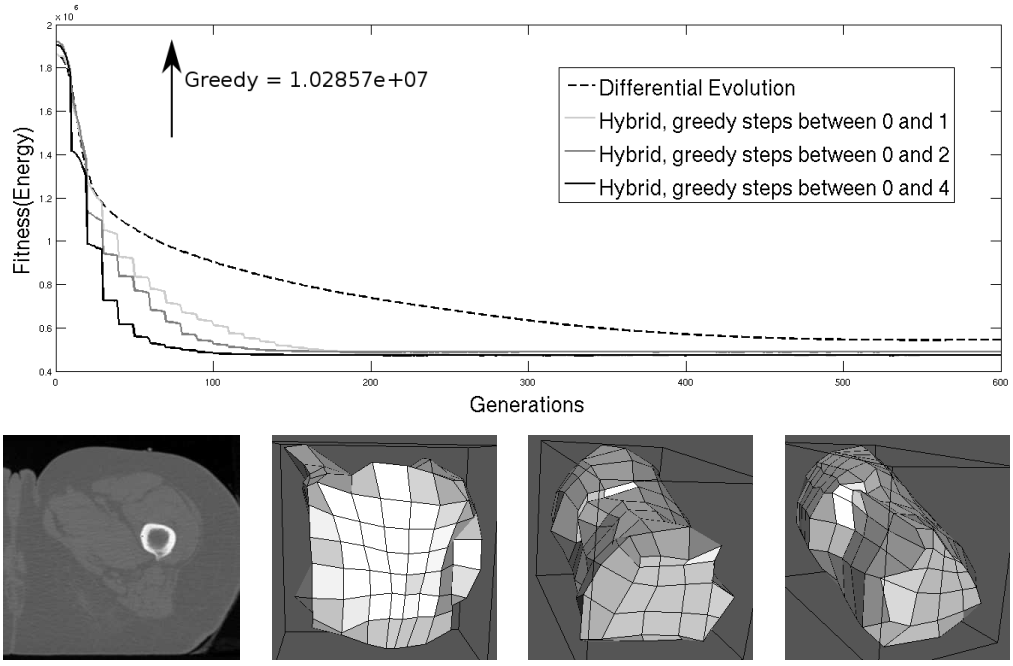
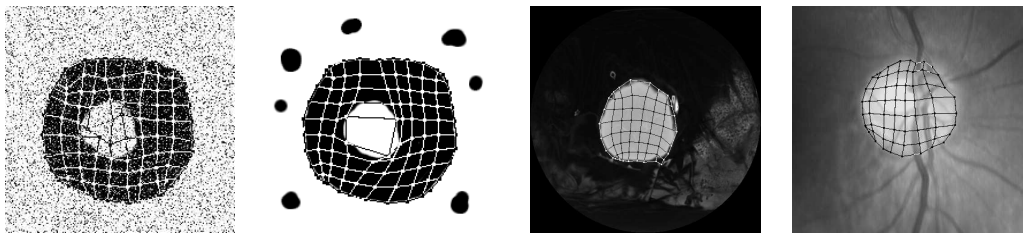
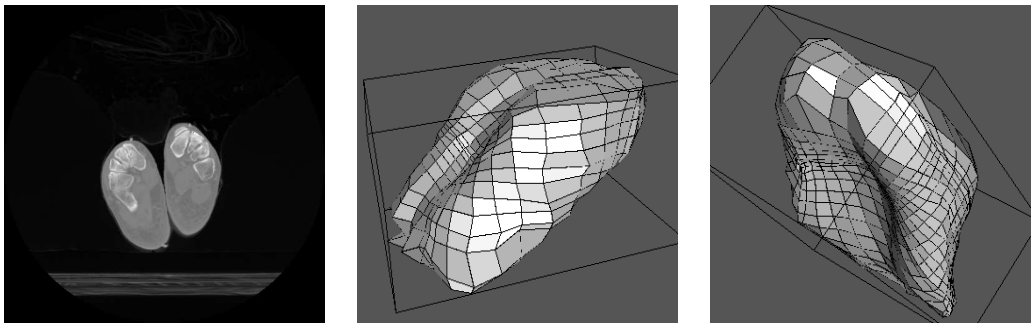


Figure 4.7: Upper graph, best individual evolution comparing the DE approach with different hybrid approaches and the greedy method in the 3D case. Bottom part, (a) CT slice of the humerus, (b) final result with the greedy method, (c) final result with DE, (d) final result with the hybridized DE, greedy steps between 0 and 4.



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

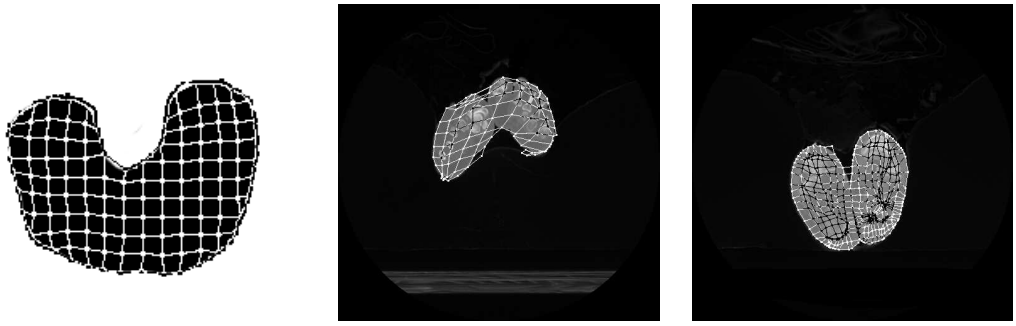
Figure 4.8: Best final results obtained using the hybridized DE in noisy 2D images. (a) Uniform Gaussian noise added. (c) Spots added as noise. (c) CT noisy image. (d) Segmentation of an optic disc.



(a) (b) (c)

Figure 4.9: Final result obtained using the hybridized DE in a CT noisy 3D image. (a) Slice of the CT image of feet. (b)(c) Different points of view of the optimized mesh.

Figures 4.8 and 4.9 show different representative segmentation examples with a high level of difficulty. In this case, we used the depicted hybrid method, applying a random number of greedy steps between 0 and 4 to all the individuals each 10 generations. In Figure 4.8, the 1^{st} and 2^{nd} examples are images with uniform Gaussian noise and spots added as local noise over an artificial image with a hole, the 3^{rd} is an original CT image of the skull with real noise produced by the scanner in the bone contour and the 4^{th} is a segmentation of the optic disc in a retinal image. Moreover, Figure 4.9 shows an example of segmentation of a real 3D noisy image. It consists of a medical CT image containing two feet. The greedy search was not able to segment these images, whereas the hybrid method overcame the noise problems providing correct segmentations.



(a) (b) (c)

Figure 4.10: 2D image segmentations that require topological changes.

4.5.1 Segmentations that require topological changes

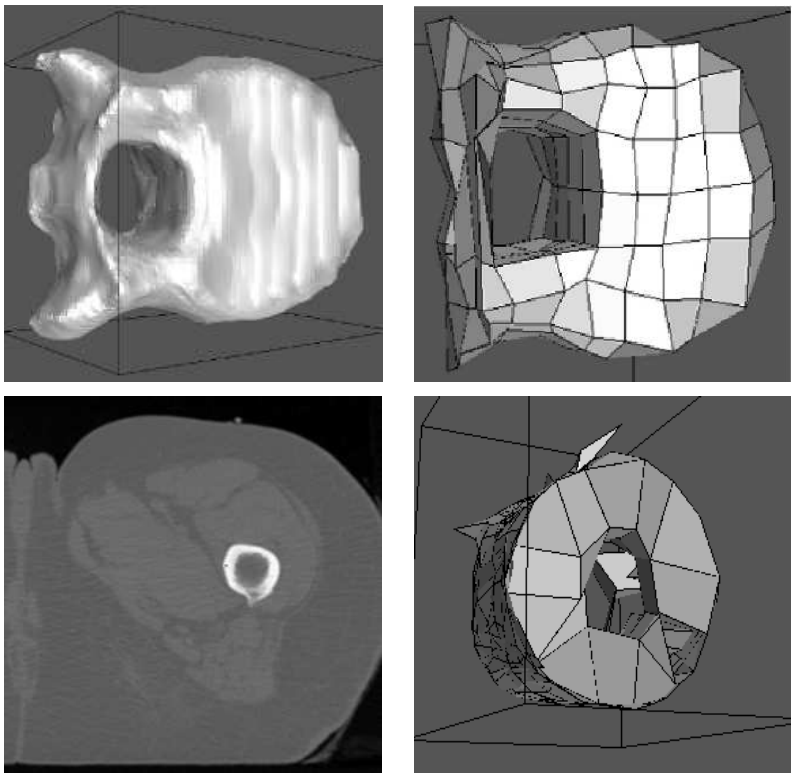
We included the possibility of topological changes provided by the greedy method, as explained previously in Section 2.3.1. Hence, the models present the capability of a better adjustment to complex objects with holes or concavities. Examples in the 2D case are shown in Figure 4.10, with an artificial image and with two CT real images, in particular two slices of two feet. The segmentation process with the cut of the links in the deep concavities was performed correctly even with the high levels of noise in such CT images.

Regarding 3D images, Figure 4.11 shows different representative medical examples with objects that require topological changes. Figure 4.11, 1^{st} row, shows the segmentation of a vertebra meanwhile Figure 4.11, 2^{nd} row, the segmentation of a humerus, both composed by CT slices. In both cases, the link cutting allows the delimitation of the hole of the segmented objects. We can point out the correct detection of the internal hole of the bone, even in this image where the bone is surrounded by flesh.

4.5.2 Segmentations that require the division of the mesh

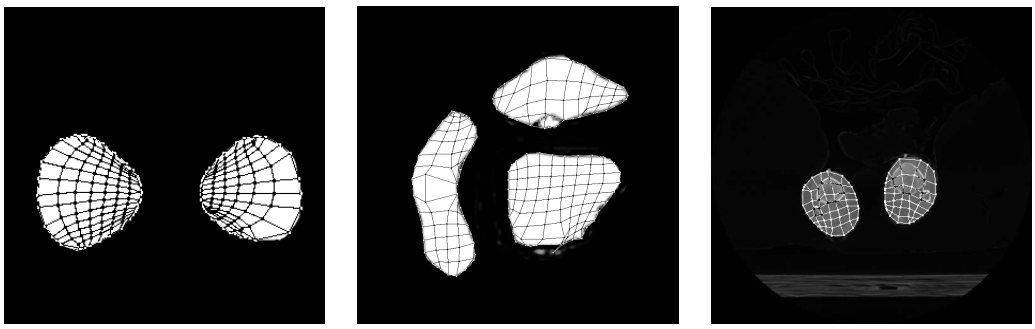
The other case that requires topological changes is the automatic division of the initial mesh. The mechanism, associated with the link cutting procedure, was explained in Section 2.3.2, and incorporated in the hybridized DE (Section 4.4). In this case, we used a random value for the F parameter in DE between 0.2 and 0.4. We reduced the maximum value for F with the aim of moving the nodes slower, avoiding the possibility of ignoring any possible object in the scene.

Thus, in Figures 4.12 and 4.13, different segmentation examples in 2D and 3D are shown. These are different artificial and real examples taken from the medical



(a) (b)

Figure 4.11: 3D image segmentations that require topological changes. (a) Original image. (b) Final results.



(a) (b) (c)

Figure 4.12: 2D image segmentations that require division of the mesh.

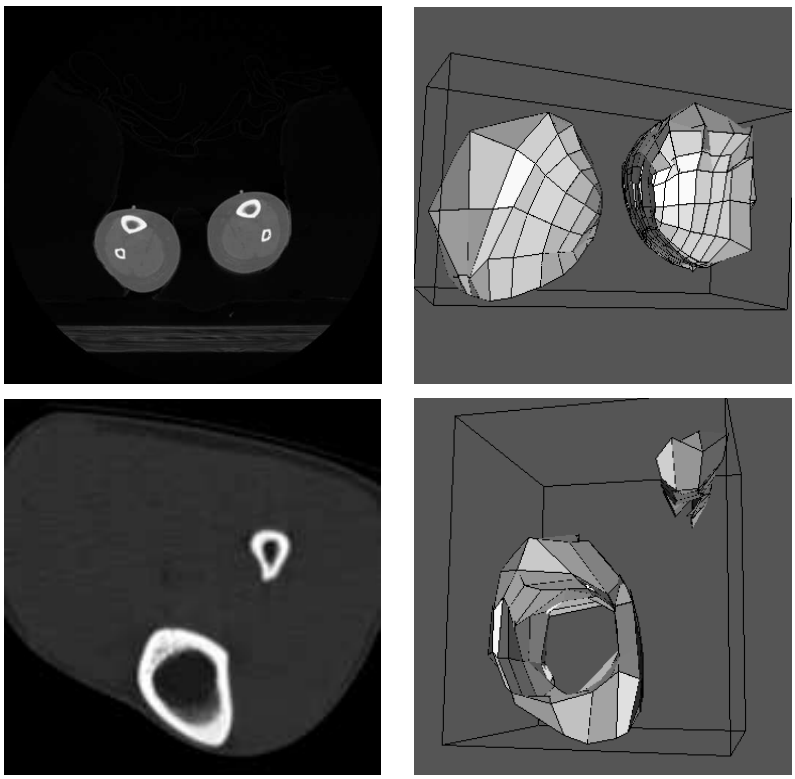
domain which present different levels of noise (especially the CT images). In all the situations the method provided correct results. Note that the segmentation overcomes the noise problem of the external contour of the flesh in the CT images as well as the high levels of noise around the bones to be segmented. In the second example of Figure 4.13 (2^{nd} row), the two procedures of the link cutting performed simultaneously the division of the 3D mesh and the delimitation of the internal hole of the largest bone, acting in different parts of the initial mesh and reaching a correct segmentation of the two objects.

Regarding computation times, the algorithms were run in an Intel Core 2 at 2.83 GHz. For example, regarding 2D and the representative example of Figure 4.6, the evolution of the DE approach across the 250 generations (1000 individuals) required an average time of 8 minutes, whereas the hybrid combination (with steps between 0 and 4) required an average time of 4 minutes to obtain the same fitness as the DE alternative (in generation 100).

Regarding 3D and Figure 4.7, the evolution of the DE approach across the 600 generations (1000 individuals used in this example) required an average of 95 minutes, whereas the hybrid combination (steps between 0 and 4) required an average time of 8 minutes to obtain the same fitness as the best value of the DE alternative (in generation 50).

4.6 Discussion

As we explained, the proposed GA approach presented some important disadvantages we tried to overcome. In this chapter, an alternative evolutionary method for the optimization of the Topological Active Model was proposed. The DE approach we implemented introduced some important advantages with respect to the classic



(a) (b)

Figure 4.13: 3D image segmentations that require division of the mesh. (a) Upper figure: 3D reconstruction of the original image; Bottom figure: 2d CT slice of the humerus. (b) Final results.

Table 4.2: Comparison between the GA and DE approaches.

	GA method	**Hybrid GA-Greedy**	**DE method**	**Hybrid DE-Greedy**
Number of genetic operators	5	5	1	1
Number of generations needed	High	Medium	Medium	Low
Topological Changes	No	Yes	No	Yes
Phases needed	2	2	1	1
Execution time	High	Medium	Medium	Medium - low

GA algorithm. Table 4.2 details the main differences of the proposed DE approaches with respect to the previous GA ones. In particular, the set of genetic operators of the GA was replaced by a single one, which implied more simplicity in order to minimize the decisions that have to be made by the designer. Moreover, the GA method required different evolutionary phases with different objectives and weights of the energy parameters. Both phases were integrated in a single one using DE. The proposed method also provided a faster convergence and better results, as it was shown in the graphics of quality evolution.

Moreover, it was developed a hybrid combination of the DE approach with the greedy local search, following the same methodology previously used with the GA method, integrating the advantages of both strategies: the global search of DE overcome the possible presence of noise in the images, whereas the greedy search helped to speed up the segmentation. Additionally, the hybrid combination introduced the possibility of topological changes to perform better adjustments and segmentations in complex surfaces or even the detection and segmentation of several objects in the scene.

Chapter 5

Evolutionary Multiobjective Optimization as an alternative approach to the optimization of the Topological Active Model

5.1 Introduction

One of the main drawbacks of the genetic algorithm was improved with the use of Differential Evolution (DE). However, the GA or DE approaches present other important drawback that should be overcome. As explained before, those evolutionary algorithms assign a fitness to each individual according to the energy associated to the model. This energy is composed by different energy components that are weighted by different energy parameters. Thus, in order to obtain correct segmentations, an experimental tuning of the weights of the energy components was necessary when a greedy local search algorithm or global search methods such as the proposed GA and DE algorithms were used in the minimization task. So, we needed to perform a parameter tuning in each segmentation process to select those ones that provided a correct result, and such tuning usually depended on the kind of image to segment.

Multiobjective Optimization Algorithms (MOAs) give a solution to this problem by considering the optimization of several objectives in parallel. The MOAs usually work with conflicting objectives trying to identify a set of optimal trade-off solutions which is called the *Pareto Set*. This set is formed by the nondominated solutions, that is, those solutions that have no other with equal values, and at least one better,

in the different objectives. Multiobjective Optimization Evolutionary Algorithms (MOEAs) [48][28] use the principles of evolutionary computing to the search of the Pareto Set. We used one of the best established algorithms of this type, the SPEA2 algorithm [110]. We tested in this chapter the improvements and problems between the use of a multiobjective optimization technique versus the use of a genetic algorithm or the DE algorithm (single-objective optimization) in our application of the optimization of the deformable model.

To our knowledge, the only work that used a MO approach to the optimization of deformable models was the work of Séguier and Cladel [83], where a multiobjective optimization of different energy components was developed to the optimization of snakes in an audio-visual speech recognition task. The authors optimized two snakes to fit the external and interior lips contours, using only a small limited number of contour points for each snake. After the evaluation of the snakes energy components, the chromosomes were ranked and, this way, the Pareto optimal solutions were searched. According to the authors, the multiobjective optimization required less iterations than an usual genetic optimization.

In the following sections we detail the main characteristics of the different evolutionary multiobjective approaches that were developed. We also show the results and comparisons to justify the advantages and improvements of the proposed methods with respect to the previous single-objective ones.

5.2 Evolutionary multiobjective optimization

Optimization problems with multiple objectives, often conflicting, arise in a natural fashion in most real-world applications. Multiobjective Optimization Algorithms (MOAs) try to identify a set of optimal trade-off solutions, the *Pareto Set*, between the conflicting objectives. If we consider k objectives to be maximized/minimized, and all are equally important, a solution to the problem can be specified in terms of a decision vector $(x_1, x_2, \ldots, x_n)$ in the decision space X. To each solution is assigned an objective vector $(y_1, y_2, \ldots, y_n)$ in the objective space Y. MOAs apply the concept of Pareto dominance to search for the solutions of the Pareto set: supposing we want to minimizes the objectives, an objective vector y^1 dominates another objective vector y^2 ($y^1 \succ y^2$) if no component of y^1 is greater than the corresponding component of y^2 and at least one component is smaller. The optimal solutions of the Pareto Front are those nondominated by any other solution, so there may exist different optimal solutions that represent different trade-offs between the objectives. Figure 5.1 shows an example of Pareto Set considering two objectives.

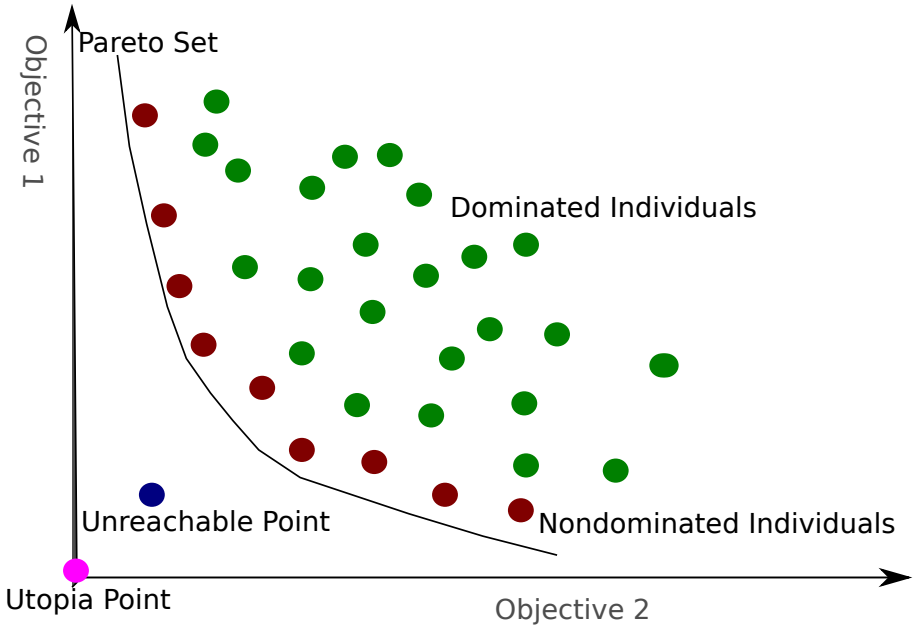


Figure 5.1: Example of Pareto Set using 2 objectives.

Multiobjective Optimization Evolutionary Algorithms (MOEAs) [48][28] use the principles of evolutionary computing to the search of the Pareto Set. As indicated by Deb [28], using evolutionary algorithms is a highly effective way of finding multiple effective solutions in a single simulation run, so several MOEAs have emerged in the last years adapted to the multiobjective optimization purpose. We used in this work one of the best established algorithms, SPEA2 [110]. It was not our purpose the comparison of the capabilities of these algorithms, as our aim was to compare the advantages and problems that add the multiobjective optimization techniques in our application.

5.2.1 The SPEA2 algorithm

SPEA2 [110] is an improved version of the *Strength Pareto Evolutionary Algorithm (SPEA)* [111]. The new version incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. Algorithm 5.2.1 includes the description of the steps of the SPEA2 procedure as described in [110].

The algorithm considers, in each generation t, a regular population P_t and an archive $\overline{P}_t$ (external set). This archive represents the set that will contain the nondominated solutions and possibly some dominated solutions if the number of

Algorithm 5.2.1: SPEA2 MAIN LOOP()

$N \leftarrow 0$ // Population size
$\overline{N} \leftarrow 0$ // Archive size
$T \leftarrow 0$ // Maximum number of generations
$A(Output) \leftarrow 0$ // Nondominated set
Step 1 : INITIALIZATION()// Generate an initial population P_0 of size N and create the empty
// archive $\overline{P}_0 = \emptyset$ of size $\overline{N}$. Set $t = 0$.
while *No Termination criterion satisfied*
do
- *Step* 2 : FITNESS ASSIGNMENT()// Calculation of fitness of P_t and $\overline{P}_t$ individuals
- *Step* 3 : ENVIRONMENTAL SELECTION()
  - // Copy all nondominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$.
  - **if** *size of* $\overline{P}_{t+1} > \overline{N}$
  - **then** *reduce* $\overline{P}_{t+1}$ *by means of the truncation operator.*
  - **if** *size of* $\overline{P}_{t+1} < \overline{N}$
  - **then** *fill* $\overline{P}_{t+1}$ *with dominated individuals in* P_t *and* $\overline{P}_t$.
- *Step* 4 : // Termination
  - **if** $t > T$(*or any other stopping criterion*)
  - **then** *Set A to the set of decision vectors represented by the nondominated individuals in* $\overline{P}_{t+1}$. STOP().
- *Step* 5 : MATING SELECTION()
  - // Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill
  - // the mating tool.
- *Step* 6 : VARIATION()
  - // Apply recombination and mutation operators to the mating pool and set P_{t+1}
  - // to the resulting population.
  - // $t \leftarrow t + 1$. Go to Step 2.

nondominated solutions is less than its size. The size of the archive ($\overline{N}$) is fixed and initially the archive is empty ($\overline{P}_0 = \emptyset$). Here we describe the main aspects of the algorithm steps:

Step 2: Fitness assignment. In this step, the fitness assignment takes into account both dominating and dominated solutions and incorporates density information into its calculation. First, each individual i is assigned a strength value:

$$S(i) = |\{j | j \epsilon P_t + \overline{P}_t \wedge i \succ j\}| \tag{5.1}$$

Where $|.|$ denotes the cardinality of the set and the symbol $\succ$ corresponds to the Pareto dominance relation. Thereafter, a raw fitness $R(i)$ is calculated based on the strength value for each individual:

$$R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j) \tag{5.2}$$

This raw fitness of an individual i is calculated using the strengths of its dominators in both the archive and population set. This use of both sets is a distinguishing difference between SPEA and SPEA2. This way, the individuals of the Pareto set have a value $R = 0$, whereas a high value of R means that the individual is dominated by many other individuals.

If the individuals have the same raw fitness values a density estimation technique is incorporated to discriminate between them. The specific estimation technique is an inverse distance of the $k - th$ nearest neighbor. This is implemented by calculating the distance of each individual to each member of the archive set and population. This distance is calculated in objective space and the results are stored in a list that is sorted in increasing order. The $k - th$ element gives the distance sought and is represented by σ_i^k, where k is equal to the square root of the sum of the population size and the archive size. Thereafter, a density $D(i)$ is then calculated using the equation:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{5.3}$$

The sum considered in the denominator is used to ensure that its value is greater than zero and $D(i) < 1$.

Finally, adding the density metric to the raw fitness, the individual fitness value is calculated as:

$$F(i) = R(i) + D(i) \tag{5.4}$$

As explained in [110], the run-time of the fitness assignment procedure is $O(M^2 logM)$, while the calculation of S and R values is of complexity $O(M^2)$, where $M = N + \overline{N}$.

Step 3: Environmental selection. This step incorporates the environmental selection, that defines how the archive maintains the Pareto Front. In contrast to SPEA, the archive set size is fixed, so SPEA2 uses a truncation method that preserves boundary points. In this step all nondominated individuals in P_t and $\overline{P}_t$ are copied to $\overline{P}_{t+1}$. The archive size constraint ($\overline{N}$), chosen by the user, is checked for $\overline{P}_{t+1}$. This is done using the truncation operator.

The archive update operation is performed as follows. First, all the nondominated individuals are copied into the archive set. If the nondominated set is equal in size to the predetermined archive set size the archive update operation is complete. Otherwise, there can be two situations: Either the archive is too big or it is too small. For the former, the best $\overline{N} - |\overline{P}_{t+1}|$ dominated individuals in the previous archive and population are copied to the new archive. For the latter case an archive truncation procedure is employed that iteratively removes individuals from $\overline{P}_{t+1}$ until $|\overline{P}_{t+1}| = \overline{N}$. At each iteration that individual i is chosen for removal for which $i \leq_d j$ for all $j \in \overline{P}_{t+1}$ with

$$\begin{aligned} i \leq_d j :\Leftrightarrow \forall 0 < k < |\overline{P}_{t+1}| : \sigma_i^k = \sigma_j^k \vee \\ \exists 0 < k < |\overline{P}_{t+1}| : [\Big(\forall 0 < l < k : \sigma_i^l = \sigma_j^l\Big) \wedge \sigma_i^k < \sigma_j^k] \end{aligned} \tag{5.5}$$

where σ_i^k is the distance of individual i to its $k-th$ nearest neighbor in $\overline{P}_{t+1}$. This procedure can be summarized as follows: The individual which has the minimum distance to another individual is chosen at each stage; if there is more than one individual with minimum distance the tie is broken by considering the second smallest distances and so on and so forth.

Steps 4-6 The termination condition check is next (Step 4). If $t > T$ (where T is the maximum number of generations) or some other stopping criteria is satisfied, then set A to the set of decision vectors represented by the nondominated individuals in $\overline{P}_{t+1}$ and then stop.

In Step 5, if the stopping condition is not met, then mating selection is performed. Binary tournament selection with replacement on $\overline{P}_{t+1}$ is used to fill the mating pool.

Finally, in Step 6, recombination and mutation operators are applied to the mating pool and set P_{t+1} to the resulting population. Increment the generation counter and go to Step 2.

5.2.2 Changes incorporated for our application

First of all, we developed the adaptation using the characteristics of the proposed GA method. Additionally, once again, we replaced the GA with the proposed DE, adaptation that will be explained in Section 5.4.

There are some simple adaptations of the previous algorithm for our application. For example, in the initialization step we used the same method applied for the GA to define the initial TAMs of the genetic population. In step 5, we incorporated the possibility of a window of individuals to perform the tournament operator instead of the binary tournament used in the original SPEA2. In step 6, we applied to the mating pool our arithmetical crossover and mutation operators, as well as the other *ad hoc* operators, that is, the whole set of genetic operators described in Chapter 3.

In addition, we refined the use of the k-distance used in SPEA2 to discriminate the individuals with the same raw fitness (step number 2). SPEA2 algorithm calculates the distance of each individual i to all the individuals in archive and population. The results are sorted in increasing order and then the $k-th$ element gives a distance σ_i^k. The authors in [110] used a value of k equal to the square root of the sum of the population size and the archive size. The value is used to finally calculate the density $D(i)$ and the final fitness value (Equations 5.3 and 5.4).

We found that this value of k is a bit arbitrary to establish the level of clustering in an area of the Pareto Front, that is, how many close neighbors an individual of the Pareto Front has. For a close group of individuals, the k value can correspond to an isolated individual so that value can deceive the aim of the density estimation technique. Instead of this, we calculated an average k-distance. First, after sorting the list of distances for each individual in increasing order, we consider only a window of the list represented by the 8th part of the closest individuals. Finally, the distance σ_i^k is an average over the k-distances that represent the 25%, 50%, 75% and 100% positions on that window. That way, the distance σ_i^k gives a more accurate view of the level of neighborhood of an individual, and with emphasis in the immediate neighborhood. Figure 5.2 shows an example of the neighborhood considered for the calculation of the k-distance of the central node.

We also changed the idea of which individuals are taken into account to calculate the distances. As we explained, the density weights the isolation level of the individual inside the population. The authors in [110] took into account the dis-

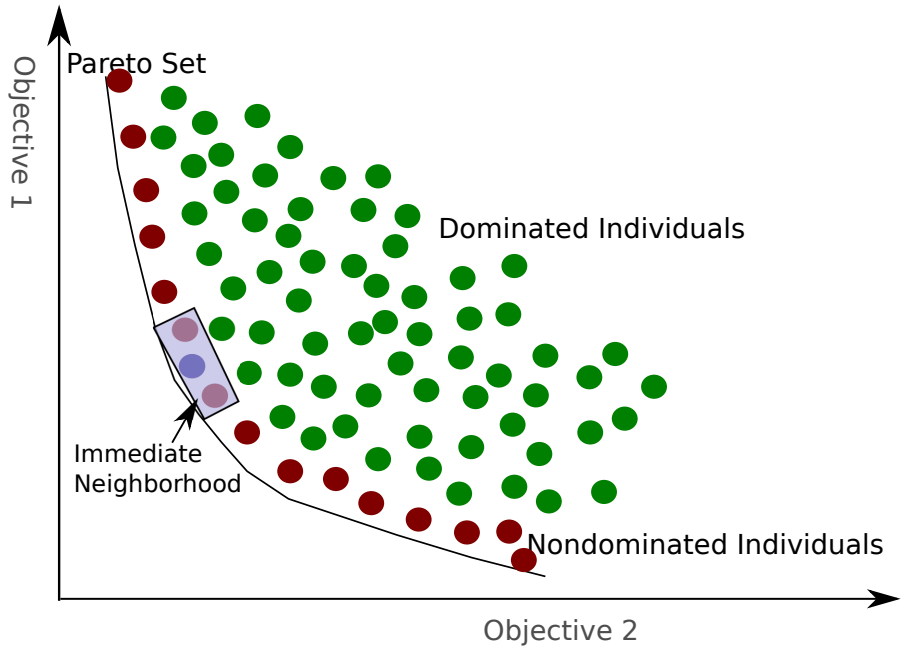


Figure 5.2: Example of the neighborhood considered for the calculation of the k-distance of the central node.

tance between each individual and the whole population, including the external set. However, we only considered the individuals from the Pareto Front to calculate this density because these are the individuals that mainly perform the archive and play the role of producing new ones for the new population. Moreover, we gained speed because we only considered the nondominated set of individuals.

Moreover, Zitzler *et al.* [110] calculated the distances in objective space, whereas we calculated the distances in the genotypic space. Zitzler *et al.* [110] applied the algorithm to benchmark functions where the intervals of the different objective functions are clearly delimited, usually the same for the different objectives. This is not the case of our application, where the values of the different objectives (energy components) can vary in a great interval and even different scales, and it is not clear how to normalize the different energy components. Nevertheless, the distances taken over the genotypic space give us the required level of clustering of individuals in a specific area of the search space and especially in the Pareto Front. Additionally, it is clear that two individuals close in the genotypic space (their coordinates) also have a similar level of neighborhood in the objective space, that is, in their energy values.

Finally, we incorporated elitism, now defined over each of the individual objec-

tives. This means that we maintain in the next generation (in the external archive) the best individuals that minimize each one of the objectives. That manner, we maintain these best individuals of the extremes of the Pareto Front. We use this option with the objectives that the user can consider as the most important to minimize, as can be the case of the gradient distance that optimizes the external contour segmentation of objects.

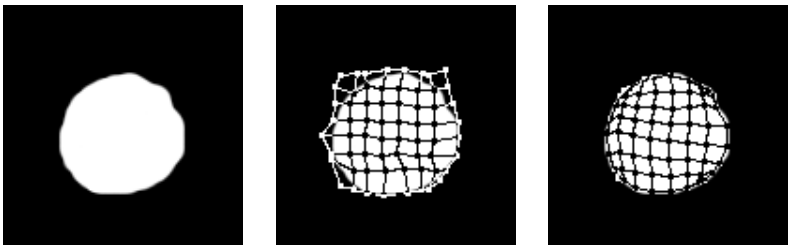
5.2.3 Evolutionary phases

In the case of the segmentations with GAs, as we mentioned, we used two evolutionary phases (avoiding the 3^{rd} that is less relevant). A first one was focused on exploration with the aim of covering or centering the individuals around the objects, and a second one focused on exploitation in order to obtain better adjustments and internal distribution of nodes. The two phases require different energy parameters, and the two parameter sets were experimentally tuned. In addition, these parameters depend on the particular image to segment.

In the case of the evolutionary multiobjective optimization methodology, and in order to obtain a better behavior in the segmentation process with the adapted SPEA2 algorithm, we defined again two evolutionary phases focused on the same specific tasks. Firstly, the population must cover the object to progressively improve the adjustment and node distribution. The main difference with respect to the GA case is that, now, the two phases are performed with the use of different objectives.

First phase. The objectives are the gradient distance (GD) and the In-Out (IO) energy terms. Therefore, we obtain individuals with the internal nodes inside the object and the external ones in the background. Moreover, the external nodes will be close to the boundaries because of the gradient distance energy term. The shift operator is only applied in this phase. We set the end of this phase in generation 100.

Second phase. Exploration predominates in the first phase while exploitation dominates this second one. The objectives now are gradient distance (GD), distance In-Out (IOD) and internal energy terms. We search for a better adjustment to the boundaries as well as a more homogeneous distribution of the internal nodes. We even add more objectives, depending on the application, as shown later. The probability of application of the spread operator is decreased because it is not as necessary as in the first evolutionary phase. This phase also finishes after a given number of generations.



(a) (b) (c)

Figure 5.3: Results obtained in the segmentation of a circle. (a) Image to segment. (b) Pareto Front individual at the end of the first evolutionary phase. (c) Pareto Front individual at the end of the second evolutionary phase.

In some segmentations, we synthesized the two external objectives in only one, called IOD_i/GD, that is, summing both energy terms. With the sum of both energy terms we try to locate the internal nodes in object intensities and the external nodes in the edges of the object in the scene. Thus, we synthesize in only one objective the desired behavior of the all the nodes with respect to the extraction of the objects. Moreover, we gain simplicity reducing one objective in the Pareto Front.

In Figure 5.3 we can see the results obtained with both phases. Figure 5.3(b) shows one of the nondominated individuals at the end of the first evolutionary phase and 5.3(c) shows one of the nondominated individuals at the end of the second phase (generation 5000, population size = 1000 individuals). These individuals were chosen from the middle area of the Pareto Front only to show the purpose of the two evolutionary phases.

5.3 Results obtained with the classic multiobjective approach

This section presents some representative segmentation examples using different images. We used for the segmentations the GA developed and the adapted multiobjective algorithm previously explained. Some of the images are artificial and were designed to show the capabilities of the MO methodology. Other images were taken from the medical domain to test the process with real ones, mainly CT medical images. Firstly, we show some results and comparisons in 2D, and after that, some results in the 3D domain.

All the processes used a population of 1000 individuals in 2D and 2000 in 3D

images. The window size of the tournament selection operator in the GA processes was 3% of the population. As we mentioned, the parameters in the GA case are different in the two evolutionary phases. The parameters used in the first phase of the GA processes were always the same. The values employed are depicted in Table 5.1. These values were tuned to obtain the aim of this phase in most of the images.

For the second phase of the GA, Table 5.2 includes the TAM parameters used in the segmentation examples. The energy parameters were experimentally set as the ones in which the genetic algorithm gave the best results for each kind of image. Table 5.3 shows the probabilities of the genetic operators used in both GA phases of the segmentations processes. Also, these probabilities of the genetic operators were experimentally tuned to obtain the best possible segmentations.

In the multiobjective processes we used an archive size of 5% and 30% of the population size in the first and second phases, respectively. The tournament window size used in both phases was 3% of the archive size. The reason of the smaller size of the archive set in the first phase is because this phase uses only two objectives (and not very conflicting), so the individuals in the Pareto Front are less than in the case of the second phase, that uses a minimum of three objectives. The probabilities of the genetic operators, in both phases of the multiobjective optimization, were the same as in the GA case. Finally, in the second phase, we used the explained elitism of two individual objectives: the gradient distance and the distance In-Out energy terms (Section 5.2.2). This means that the individuals with the best minimum values in those terms are maintained in the archive.

In the case of 3D segmentations, the input to the algorithms is a set of consecutive slices with such properties. In the 3D examples we used between 90 and 200 slices in the different examples. We synthesized the two external objectives in only one called IOD_i/GD, that is, summing both energy terms. With this new objective, we try to locate the internal nodes in object intensities and the external nodes in the contour of the object. Moreover, we gain simplicity reducing one objective. Regarding the internal energy, from the 3 energy terms (α, β and γ), we picked only the β energy term, that is, the one that provides smoothness to the mesh.

All the results obtained with the GA or the multiobjective optimization were taken about generation 5000, given the complexity that is necessary for progressively improving a Pareto Front given the different objectives. Using a bigger number of generations did not provide significant improvements.

Table 5.1: TAM parameter set of the GA first phase in the segmentation processes of the examples. (γ only used in 3D segmentations).

α	β	γ	ω	ρ	ξ	δ	τ
0.00001	0.00001	0.0	1.0	1.0	0.0	10.0	0.0

Table 5.2: TAM parameter sets of the GA second phase in the segmentation processes of the examples (γ only used in 3D segmentations).

Figure	*Size*	α	β	γ	ω	ρ	ξ	δ	τ
5.3	8×8	3.0	1.5	-	1.0	4.0	5.0	10.0	1.0
5.5 1^{st} row	10×10	3.5	1.5	-	1.0	4.0	5.0	10.0	1.0
5.5 2^{nd} row	10×10	4.0	1.0	-	1.0	4.0	5.0	8.0	1.0
5.5 3^{rd} row	10×8	3.0	1.5	-	1.0	4.0	5.0	10.0	2.0
5.5 4^{th} row	10×8	3.5	2.0	-	1.0	3.0	5.0	8.0	2.0
5.7 1^{st} row (b)	10×10	4.0	2.0	-	1.0	4.0	5.0	8.0	1.0
5.7 1^{st} row (c)	10×10	4.0	2.0	-	10.0	4.0	5.0	8.0	50.0
5.7 2^{nd} row (b)	12×12	3.5	2.0	-	1.0	4.0	5.0	10.0	1.0
5.7 2^{nd} row (c)	12×12	3.5	2.0	-	10.0	4.0	5.0	10.0	10.0
5.8 2^{nd} row (a)	$7 \times 7 \times 5$	2.0	1.5	1.0	1.0	4.5	5.0	10.0	1.0
5.8 3^{rd} row (a)	$7 \times 7 \times 5$	2.0	1.5	1.0	10.0	4.5	5.0	10.0	10.0
5.8 2^{nd} row (b)	$9 \times 9 \times 6$	2.0	1.5	0.5	1.0	4.5	5.0	10.0	1.0
5.8 3^{rd} row (b)	$9 \times 9 \times 6$	2.0	1.5	0.5	10.0	4.5	5.0	10.0	10.0
5.9 1^{st} row (b)	12×12	3.5	2.0	-	1.0	4.0	5.0	10.0	1.0
5.9 1^{st} row (c)	12×12	3.5	1.5	-	10.0	4.0	5.0	10.0	10.0
5.9 2^{nd} row (b)	12×12	3.5	2.0	-	1.0	4.0	5.0	10.0	1.0
5.9 2^{nd} row (c)	12×12	3.5	2.0	-	10.0	4.0	5.0	10.0	10.0

Table 5.3: Genetic operators probabilities used in the evolutionary processes.

Phase	*p.crossover*	*p.mutation*	*p.spread*	*p.shift*	*p.group mutation*
1	0.5	0.001	0.01	0.05	0.002
2	0.5	0.005	0.005	0.0	0.01

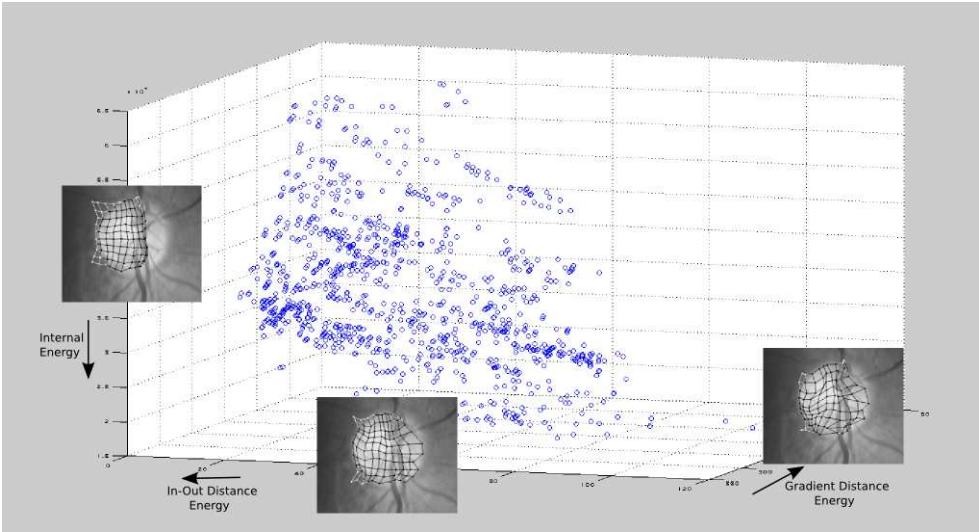


Figure 5.4: Representation of the Pareto Front in a segmentation process, in a generation of the second evolutionary phase.

5.3.1 Analysis of the Pareto Front

One of the main advantages of the evolutionary multiobjective optimization is that it provides an entire set of possible solutions for our segmentation task depending on the different trade-offs among the objectives considered. These possible solutions are the members of the Pareto Front. In Figure 5.4, we show the 3D representation of the Pareto Front in a 2D segmentation process of a retinal image. This representation was taken from a generation (3000) of the second phase, when we considered 3 objectives: gradient distance, distance In-Out and internal energy terms. In this example, the two internal energy terms (Equation 2.2) were considered as a summed objective ($\alpha = 1$, $\beta = 1$). In most cases we considered the internal energies as only one objective, because the relevance of each energy component (contraction and bending) is equally important. Additionally, we do not increment the number of individual objectives the algorithm must consider. In the figure it is also shown the individual of the Pareto Front that minimizes each of the three objectives. We used elitism of the best individual regarding gradient distance and distance In-Out objectives and the averaged k to calculate the distances, as previously explained.

5.3.2 Segmentation results obtained in artificial images

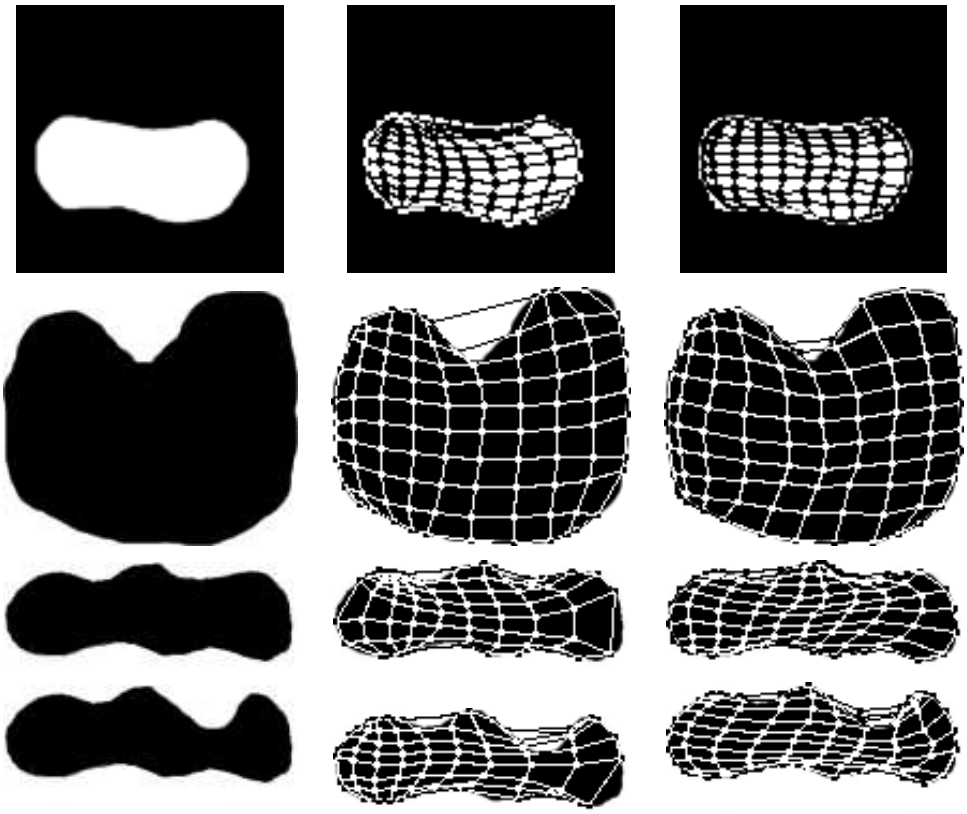
Firstly, we tested the methodology with some artificial images. We present 2D segmentations in Figure 5.5, where some different examples are shown. The results

shown in column (b) correspond to the GA case and those of column (c) correspond to the multiobjective approach. In all the results with the multiobjective optimization we can point out the correct boundary detection as well as an homogeneous distribution of the nodes.

The results of column (b), in second, third and fourth rows in Figure 5.5, with the classic GA methodology developed, indicate that the GA method have some problems in dealing with stretched objects. In the second row, the GA performed a correct segmentation, but the multiobjective algorithm obtained a correct segmentation with a more homogeneous distribution of the external nodes, which provided a better detection of the concavity. In the third and fourth rows, the GA falled in local minima with a not very homogeneous distribution of the internal nodes, together with the previous problem in concavities.

With the multiobjective optimization we can extract individuals from the Pareto Front that performed correct segmentations in these special situations. The results obtained in Figure 5.5, for the multiobjective processes, were selected as the individuals from the Pareto Front that minimized the global summed energy -according to the parameter set used in the corresponding GA process-. It is interesting to highlight that, although the nondominated individuals were chosen with the same minimization criterion of the GA (the parameter set), these nondominated individuals have better values in the different objectives, as it can be visually guessed. The reason is the parallel progression of the Pareto Front to progressively better individuals with better individual energy components.

Regarding 3D images, firstly, we tested the method with some artificial images. In Figure 5.6 the results obtained with different images are shown. The 1^{st} row shows the original image to be segmented or a composing slice (first column), the 3^{rd} row shows an intermediate individual taken from the Pareto Front and 2^{nd} and 4^{th} rows correspond to the extremes of the Pareto Front for objectives β and IOD_i/GD, respectively. These meshes shown in Figure 5.6, 2^{nd} row, correspond to the best segmentations regarding the β objective (smoothness), with the optimum values (best possible smoothness), because the procedure creates the initial random TAVs with cubic and regular distributions of the nodes. Nevertheless, although the best individual regarding such objective is present from the beginning, it acts as a point that delimits and forces the progression of the trade-offs nondominated individuals of the Pareto Front. As we can see in the results, the individual that optimizes the IOD_i/GD objective, provided the best adjustment to the contour that the process could obtain. In all the trade-offs obtained from the middle of the Pareto Front. we can point out the compromise obtained between the smoothness and the contour



(a) (b) (c)

Figure 5.5: Results obtained in the segmentation of different stretched objects. (a) Image to segment. (b) Result obtained with the GA method. (c) Examples of nondominated individuals at the end of the segmentation process.

detection searched in the extremes of the Pareto Front.

We also tested the algorithm to segment images with objects with a high level of complexity, in particular objects with deep concavities and holes. Regarding 2D segmentations, in Figure 5.7, we can see two segmentation examples. The GA method, with a usual set of parameters, cannot obtain a good detection of the holes and concavities (Figure 5.7 (b)). This parameter set corresponds to the intervals of the tuned parameters for not very complex images as the previous ones of Figure 5.3 and 5.5. In this case, we had to perform a special tuning of the parameters to detect them (Figure 5.7 (c)), forcing the weight parameters ω and especially τ to high values (Equations 2.6 and 2.8), in order to obtain acceptable segmentations.

Meanwhile, the multiobjective process provided several good TAN segmentations because of the IOD acting as an individual objective. It can be seen in Figure 5.7 (d). These individuals in column (d) were selected from the nondominated individuals of the Pareto Front as the ones that minimized the IOD objective. Although they minimize this particular objective, they maintain a correct adjustment to the object boundaries as well as a homogeneous distribution of nodes (the other objectives). This is again because of the evolution of the Pareto Front to individuals that progressively improve all the objectives.

Regarding 3D segmentation, we also included some results to show the same problem of a single-objective optimization procedure. The GA method, using a typical parameter configuration, cannot obtain a correct detection of the holes and concavities. This is shown, for example, in Figure 5.8, 2^{nd} row, where some internal nodes fall in the hole, which is not correctly delimited. In this case, it is necessary to perform a specific tuning of the parameters to detect them (forcing the weight parameters ω and τ to high values), as shown in Figure 5.8, 3^{rd} row. However, the multiobjective method, due to the IOD_i/GD objective, preserved individuals of the Pareto Front that perform a correct detection in these complex areas. This can be seen in Figure 5.8, 4^{th} row, representing the best individual from the Pareto Front considering the mentioned objective, IOD_i/GD.

5.3.3 Segmentation results obtained in real images

We used CT images taken from the medical domain. Thus, we can analyze the segmentation results in a real domain. Regarding 2D segmentations, in Figure 5.9 two representative examples are shown. The images correspond to CT slices of the leg. In these images the internal holes indicate the bone area. Once again, the GA method had many problems to perform a correct detection of the internal holes (5.9(b)) with the use of typical energy parameters, so we had to take a special set of

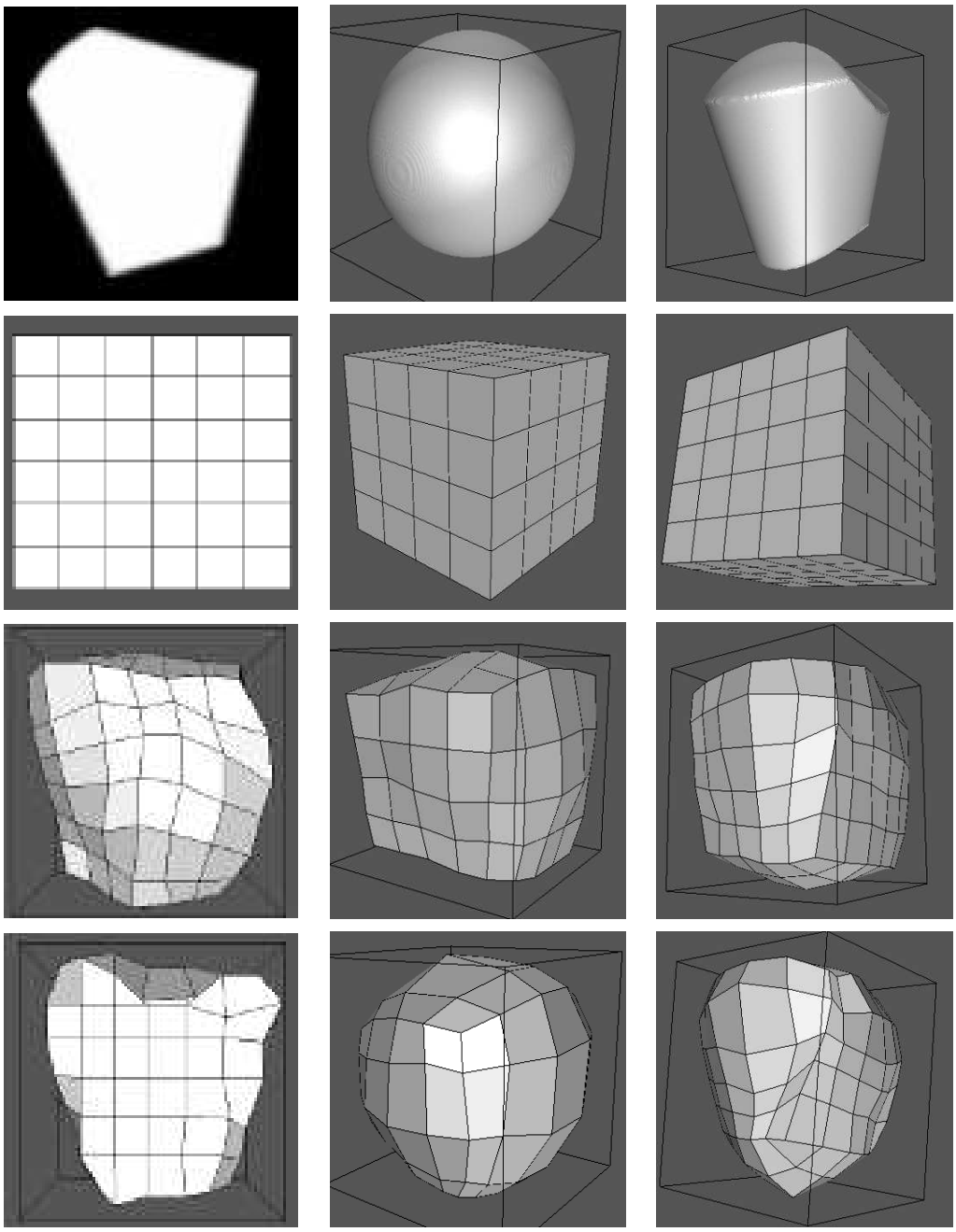
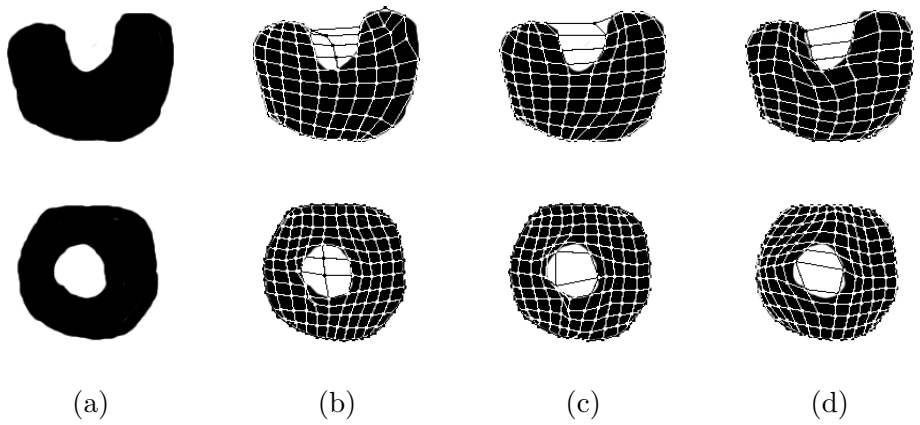


Figure 5.6: Results obtained in the segmentation of artificial images. 1^{st} row, original image. 2^{nd} row, best individual by β objective. 3^{rd} row, intermediate nondominated individual. 4^{th} row, best individual by IOD_i/GD objective.



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

Figure 5.7: Results obtained in the segmentation of complex objects. (a) Image to segment. (b) Final results with the GA method and usual parameters. (c) Final results with the GA method and specific parameters for these images. (d) Nondominated individuals at the end of the evolutionary multiobjective process (with the best IOD objective).

parameters to be able to detect this kind of internal holes (5.9(c)). On the contrary, the multiobjective optimization approach presents several individuals that perform an adequate detection of these internal holes. The ones shown in Figure 5.9(d) were selected as they minimize the IOD objective. As in the previous case, these segmentations present correct boundary detections and distribution of nodes.

Regarding 3D segmentations, we also tested the proposed method in CT images. In Figure 5.10 three representative examples are shown presenting a high level of complexity or a significant level of noise surrounding the surface of the objects. The proposed method provided a correct segmentation of the objects in these difficult situations. The nondominated individuals correspond to those which minimize the IOD_i/GD objective. Figure 5.10, 1^{st} row, corresponds to a foot. The input images are noisy CT images of different slices of such foot. Due to the complex surface of this object, we performed a previous stage in this segmentation to obtain a population of individuals that firstly identify the boundary of the foot.

The second example corresponds to the reconstruction of a vertebra, taken from CT slices. The selected nondominated individual delimits correctly the internal hole, performing a reasonable boundary detection. The third one corresponds to the segmentation of a humerus. In this case, the CT images are very noisy, with an additional external contour (the flesh of the leg) that the optimization method must avoid. Moreover, the boundary of the bone is fuzzy. This can be seen in the final nondominated and selected individual, where a group of external nodes stretch the mesh in a extreme to delimit the brighter area of the bone boundary, as the CT

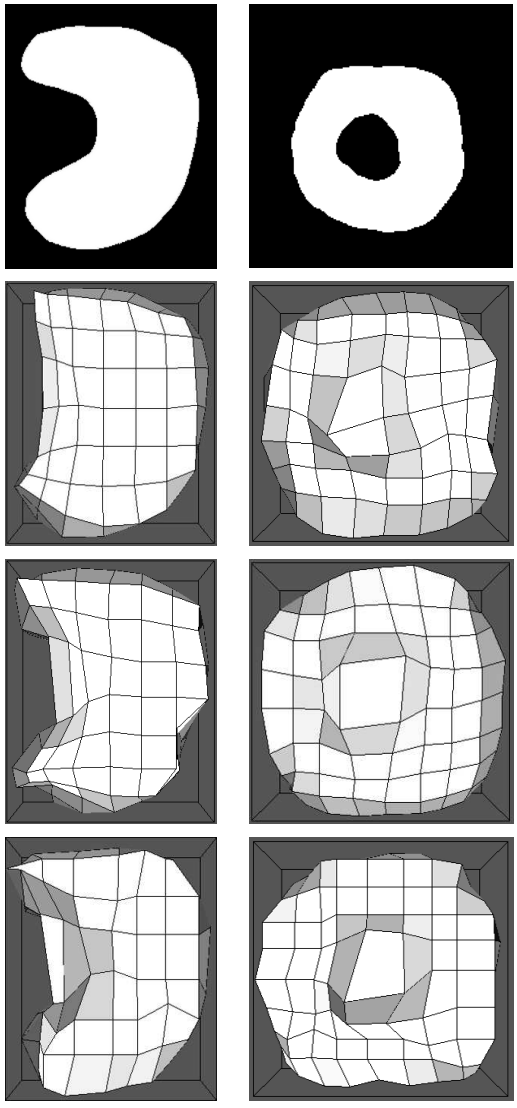
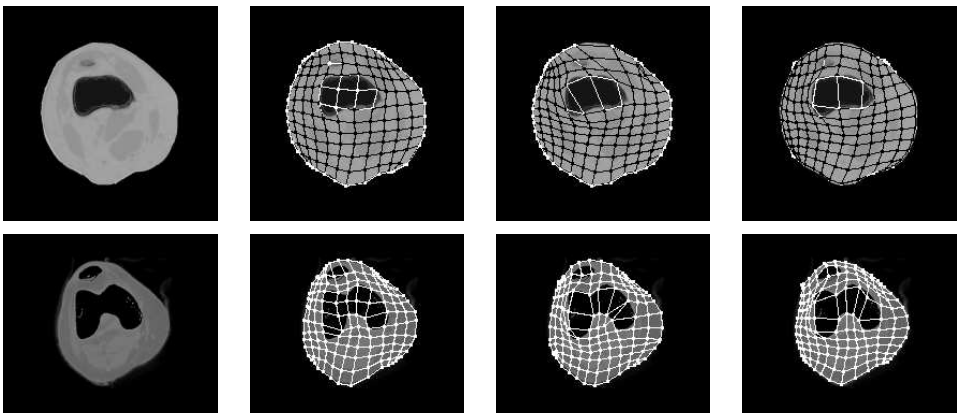


Figure 5.8: Results obtained in the segmentation of complex objects. 1^{st} row, slice of the original image. 2^{nd} row, final results with the GA method and typical parameters. 3^{rd} row, final results with the GA method and specific parameters. 4^{th} row, nondominated individuals at the end of the evolutionary process.



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

Figure 5.9: Results obtained in segmentations with CT images. (a) Image to segment. (b) Final results with the GA method and usual parameters. (c) Final results with the GA method and specific parameters for these images. (d) Examples of final Pareto Front individuals that minimize the distance In-Out with the multiobjective approach.

images of such extreme indicate.

5.4 The adapted SPEA2 algorithm combined with Differential Evolution

As we proved in Chapter 4, Differential Evolution (DE) [77, 78] introduces more simplicity in the evolutionary process, because it replaces the production of new individuals, using the whole set of GA operators, with the single operator of Differential Evolution. Moreover, with the DE approach, we obtained a faster convergence of the population over the desired results.

As we detailed in Section 5.2.2, we used the characteristics of the proposed GA to implement steps 5 and 6 of the SPEA2 algorithm (Algorithm 5.2.1). Following the same idea, we modified the implementation of the SPEA2 algorithm to introduce DE in the production of new individuals. Thus, in Algorithm 5.4.1, the change introduced in the SPEA2 main loop is shown, replacing previous steps 5 and 6 by this new one. The rest of steps remains the same.

Step 5 We introduced the DE approach in the multiobjective algorithm. In the basic SPEA2 algorithm (using the GA operators), if the stopping condition is not met, then mating selection was performed using binary tournament selection with replacement on $\overline{P}_{t+1}$ to fill the mating pool. After that, recombination

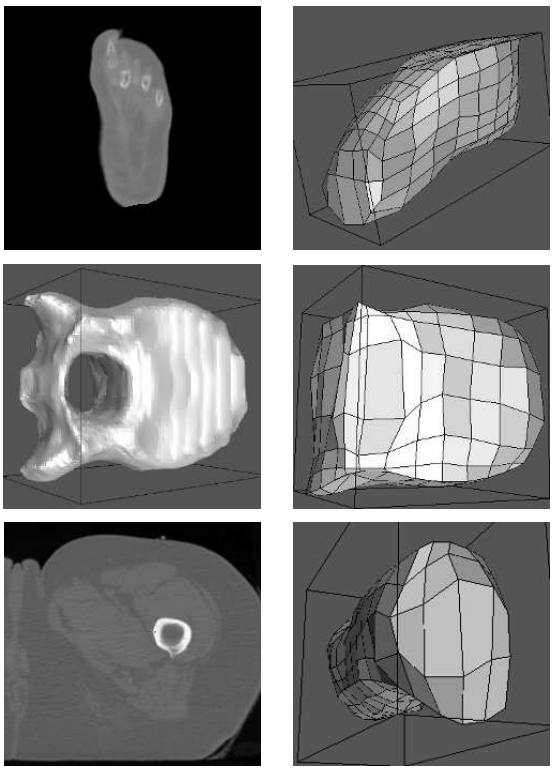


Figure 5.10: Results obtained in the segmentation with real images. First column, CT slice or 3D reconstruction of the object to segment. Second column, example of nondominated individual at the end of the evolutionary process.

Algorithm 5.4.1: MODIFICATION WITH THE HYBRIDIZED SPEA2-DE MAIN LOOP()

Step 5 : DIFFERENTIAL EVOLUTION()// Generation of next population
// Generation of new population according to DE : candidate solutions
// are defined from the individuals in $\overline{P}_{t+1}$. Go to Step 2.

and mutation operators were applied to the mating pool and set P_{t+1} to the resulting population.

In the hybridization with DE, this step defines again how the next population (P_{t+1}) is obtained. A candidate solution y is defined according to the DE algorithm (Algorithm 4.2.1), where individual x_1 is chosen by tournament only from the archive ($\overline{P}_{t+1}$), whereas x_2 and x_3 are randomly chosen from the entire population.

Moreover, in the DE method, the candidate solution replaces the original one if it is better in terms of fitness. So, we have to deal with another question, that is, when to replace the original individual with the candidate solution. In single-objective optimization, the decision is easy: the candidate replaces the original one only when the candidate is better. In MOAs, on the other hand, the decision is not so straightforward. We could use the concept of dominance (the candidate replaces the original individual only if it dominates it), but, as indicated by Robič and Filipič [80], this would make the greedy selection scheme of DE even greedier. Because of that, in our proposed hybridization, the candidate solution is introduced directly into the new population. Note that the fitness comparison is implicitly performed by SPEA2 in Step 3, as only the best individuals (in terms of the fitness explained in Step 2), being nondominated most of them, are stored in the external archive ($\overline{P}_{t+1}$).

In this multiobjective optimization hybridization with DE, we merged the two phases in only one. We initialized the individuals of the population with a minimum size, enough to guarantee that the meshes will cover the entire object to be segmented. We will use the following energy terms as independent objectives: internal energy using the second derivative (with value of β different of 0) that provides smoothness in the mesh, and the merged IOD_i/GD objective (explained in Section 5.2.3) that provides adjustment to the object.

5.5 Results obtained with the hybridized SPEA2-DE approach

We selected some representative segmentation examples to show the capabilities and advantages of the hybridization of the Multiobjective Optimization (MO) approach with the explained DE method.

Most of the processes used a population of 1000 individuals, especially in the 2D segmentations. In the 3D segmentations, some examples used 2000 individuals. We

also required between 300 and 2000 generations in most of the 2D segmentations and between 4000 and 5000 in the case of 3D ones. Regarding the DE algorithm, the tournament size to select the base individual x_1 in the DE runs was 10% of the population. As explained in Chapter 4, we used a fixed value for the CR parameter (1.0), whereas we used a maximum value of 0.4 for the F parameter. In the different applications of the equation which determines a candidate solution (Algorithm 4.2.1), we used a random value for F between 0.2 and such maximum value (for each node), parameters that were experimentally tuned to provide the best results in most of the images.

5.5.1 Comparison between the alternatives implemented

Firstly, we performed a comparison of both methodologies, that is, the multiobjective optimization method of SPEA2 using the classic genetic operators defined for a GA and the hybridized SPEA2 using differential evolution (SPEA2-DE).

In the examples presented we used only one evolutionary phase, with a minimum approximated size of the active nets of the initial population. This guarantees that the object presented in the image can be covered by the initial meshes, avoiding the possible falls in parts of them (that is mainly the objective of the first phase previously used). Thus, it is easier to show the difference between the two ways of producing new populations in the MO algorithms, that is, the use of the classic GA operators and differential evolution.

Figure 5.11 shows a comparison of the evolution of one of the objectives (IOD_i/GD) using both implemented versions of the SPEA2 algorithm. That is, we extracted from the Pareto Set, in each generation, the individual with the best (minimum) IOD_i/GD objective. This comparison was performed using the object shown in Figure 5.12. Given the stochastic nature of evolutionary algorithms, the graphs are an average of different runs (20) with different initial populations. Figure 5.13 shows the same comparison in the 3D case, and using the object in Figure 5.14. Both comparisons used the initializations of individuals (TANs and TAVs) explained before. We selected the IOD_i/GD objective to show the comparison because is the most relevant one, that is, the objective that represents the best adjustment to the object in the scene.

Thanks to the inclusion of the DE technique in the production of new individuals, we simplified the evolutionary process because the set of genetic operators (crossover, mutation, group mutation, spread and shift of the GA), together with their corresponding frequencies of application, were avoided. Moreover, we obtained a faster convergence. This can be seen in Figures 5.11 and 5.13, with the faster evolu-

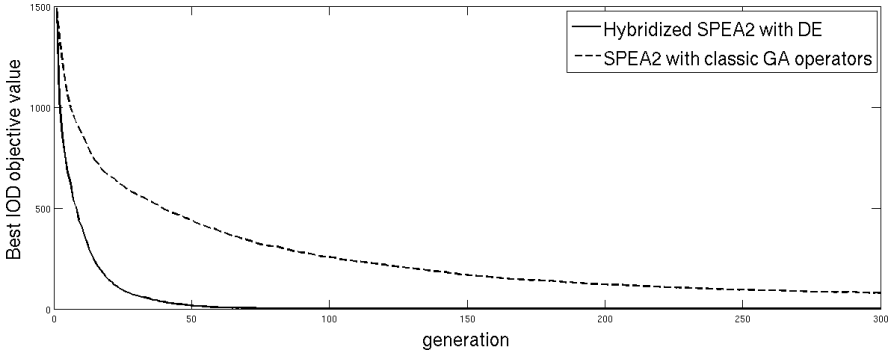


Figure 5.11: Best value of the IOD_i/GD objective of the population with the different MO processes in 2D. The curves are an average of 20 different runs with different initial populations.

tion of the IOD_i/GD objective across generations using the hybridized SPEA2-DE with respect to the basic SPEA2 algorithm (with the classic genetic operators). The faster segmentation is also shown in Figures 5.12 and 5.14, where we can see the faster adjustment of the external nodes to the external contour of the objects.

Moreover, we performed a comparison between the hybridized method and the basic DE procedure (single-objective) to emphasize the advantage of the multiobjective methodology. In Figure 5.15 the results with both methodologies are shown using a representative example, a CT slice of a knee. Note that this CT image has a noisy neighborhood in the surrounding of the object contour we want to segment (the flesh of the leg) and two internal holes that correspond to the internal bones of the knee. The first image (Figure 5.15, a) is the final result obtained using the single-objective DE approach (using 1000 individuals and 300 generations). We had to establish specific energy parameters (that define the summed energy or fitness) to obtain a correct detection of the inner hole, parameters that were experimentally tuned. The energy parameters used are depicted in Table 5.4.

Meanwhile, with the hybridized MO approach, we could extract different final segmentations. In particular, in this Figure, the best individuals according the IOD_i/GD and β objectives are shown (after 500 generations using 1000 individuals in the evolutionary population). In the minimization of the first one we obtained the best external adjustment to the object (Figure 5.15, b), meanwhile with the minimization of the second one we obtained individuals with the smoothest distribution of nodes (Figure 5.15, c). The best by the β objective is not perfectly squared (ideally the best possible smoothness) because we only introduced elitism over the IOD_i/GD objective, so the initial squared individuals were lost over the genera-

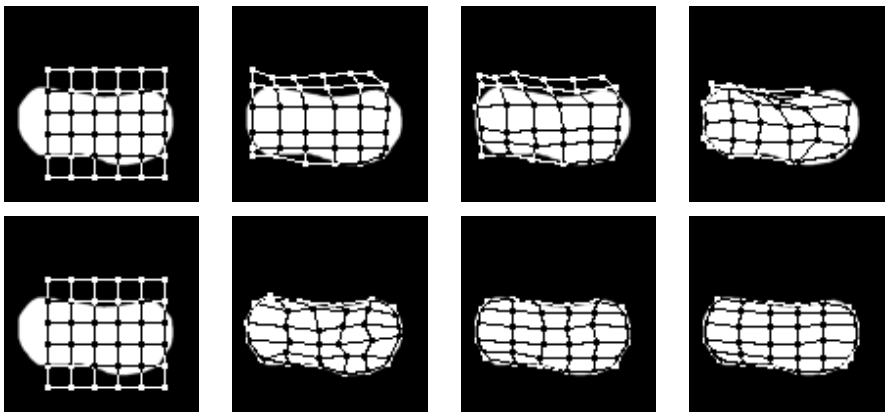


Figure 5.12: 2D sequence of the best individual (according IOD_i/GD objective) in generations 0, 50, 100 and 300. 1^{st} row, classic SPEA2. 2^{nd} row, hybridized SPEA2 with DE.

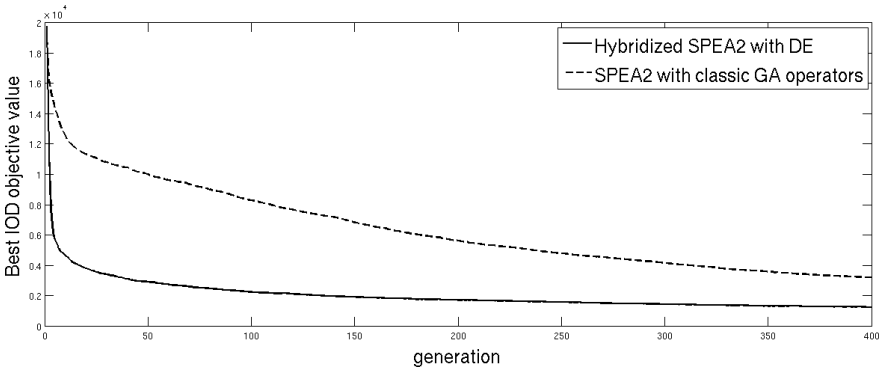
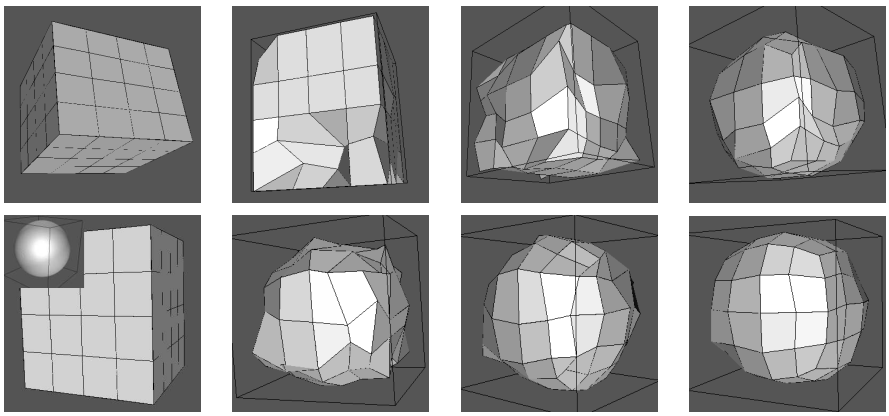


Figure 5.13: Best value of the IOD_i/GD objective of the population with the different MO processes in 3D. The curves are an average of 20 different runs with different initial populations.



Gen. 0 Gen. 50 Gen. 400 Gen. 4000

Figure 5.14: 3D sequence of the best individual of the Pareto Set according the IOD_i/GD objective. The figures correspond to generations 0, 50, 400 and final generation (4000). 1^{st} row, classic SPEA2. 2^{nd} row, hybridized SPEA2 with DE. The inset in 1^{st} figure of 2^{nd} row is an artificial reconstruction of the object.

Table 5.4: TAN parameter set used for the segmentation of the image of Figure 5.15, using DE (single-objective).

Figure	*Size*	α	β	ω	ρ	ξ	δ	τ
5.15 (a)	8×8	2.0	3.5	10.0	4.0	5.0	6.0	30.0

tions. Note that, although the best β objective corresponds to the perfect squared meshes, its use as an independent objective allows the search of the diversity of non-dominated individuals between the two basic objectives. One of these intermediate individuals is also extracted (Figure 5.15, d), taking a correct external adjustment and also a smooth node distribution, thanks to the simultaneous minimization of both objectives in the progression of the Pareto Front across the evolutionary generations. This "intermediate" solution taken from the Pareto Front corresponds to the individual of the Pareto Front that minimizes a single-objective energy, using the same weights or energy parameters of Table 5.4. Note that, once we obtain the final Pareto Front, we could select other different individuals corresponding to different criteria.

Regarding computation times, the algorithms used in Figure 5.15 were run in an Intel Core 2 at 2.83 GHz. The hybridized SPEA2-DE took an average time of 12 minutes to complete the 500 generations necessary to achieve the results shown in Figures 5.15 (b),(c) and (d), meanwhile the DE approach needed between 6 and 7

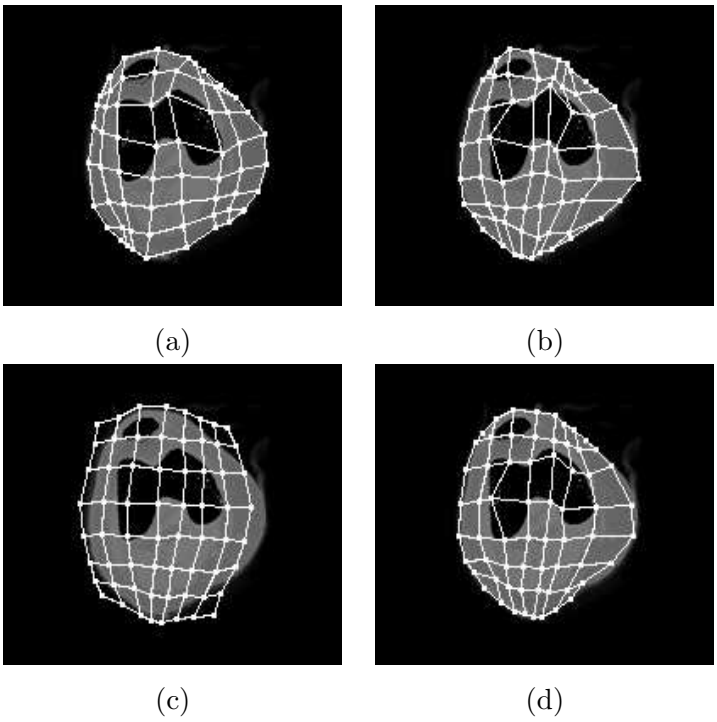


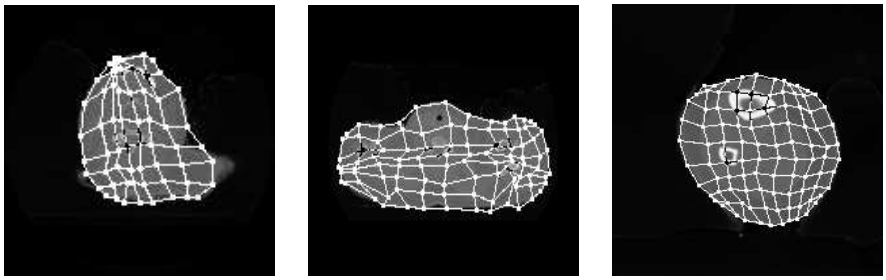
Figure 5.15: Results of the comparison between the hybridized SPEA2-DE and single-objective DE using a CT image of the knee. (a), Result obtained using DE with a particular parameter set. (b), (c) and (d), individuals taken from the final Pareto Set: best by IOD_i/GD, best by β and a final active model taken from the middle of the Pareto Front, respectively.

minutes to reach the final result shown in Figure 5.15 (a), ending in approximately the 300 generations applied. This is the main disadvantage of the MO approach with respect to the single-objective optimization using DE alone, given the computational complexity of the former (commented in Section 5.2). Nevertheless, as we are working with off-line segmentations, the computational times in both methodologies are acceptable.

5.5.2 Results of the hybridized multiobjective method with differential evolution

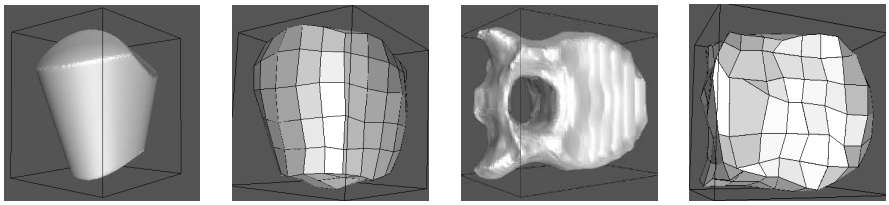
Finally, to show the capabilities of the hybridized approach in more detail, in this section different representative examples in 2D and 3D domain are presented. In Figure 5.16, the examples correspond to some CT slices from different parts of the body: a slice of the head, of the body (at the level of the shoulders) and a leg, respectively. All the images present a significant level of noise surrounding the objects, produced by the CT scanner, which complicates the detection of the contour. Moreover, the contour of the objects to be segmented present different levels of complexity, such as a simple round contour of the leg (Figure 5.16, c), a stretched one as in the body at the level of the shoulders (Figure 5.16, b) and the difficult contour of Figure 5.16 (a), which includes a small part of one shoulder, which makes more complex a correct detection of all the flesh. The topological active nets of these figures correspond to the individuals of the Pareto Front that minimize IOD_i/GD, obtained using 1000 individuals after 2000 generations.

Moreover, we also tested the methodology in 3D images. In Figures 5.17 and 5.18 representative segmentations are shown. The examples in Figure 5.17 are an artificial object and a vertebra. The artificial object was created as the intersection of a couple of cones that configure a good example to test the method, with smooth and rough parts in the contour. The vertebra was defined from CT slices that were preprocessed to isolate the bone, configuring an object to segment with a complex surface, including a hole. The final segmentations in the Figures correspond to the individuals of the final Pareto Front that minimize again the IOD_i/GD objective, using a total number of 2000 individuals and 4000 generations, in Figure 5.17, and 5000 generations in Figure 5.18. In both cases, the best individual that minimizes the IOD_i/GD objective provided correct segmentations. Moreover, Figure 5.18 presents the segmentation of the humerus in medical CT images. This example presents a high complexity because the bone is surrounded by the flex and has an internal hole so it has a high level of noise. These complications have to be overcome to reach the bone contour. Despite the noise, as a global search method, the SPEA2-



(a) (b) (c)

Figure 5.16: Results obtained in the segmentation of 2D examples. (a) CT slice of the head, (b) CT slice of the body, at the level of the shoulders, (c) CT slice of a leg. The TANs correspond to individuals taken from the Pareto Set that minimize the IOD_i/GD objective.



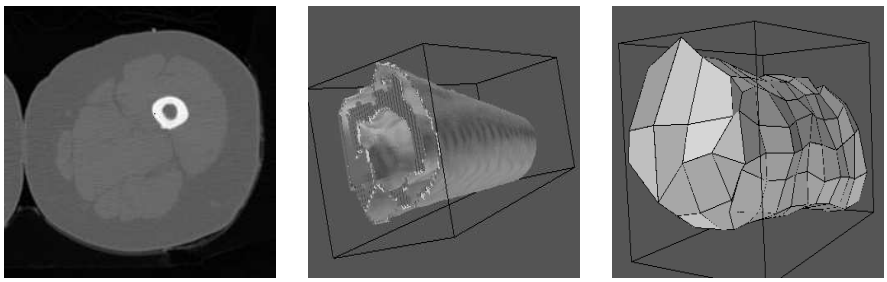
(a) (b) (c) (d)

Figure 5.17: Results obtained in the segmentation of 3D examples. (a) Snapshot of the artificial object. (b) Best result according the IOD_i/GD objective. (c) 3D reconstruction of the vertebra composed of 2D CT slices. (d) Best final result according the minimization of the IOD_i/GD objective.

DE approach overcomes the difficulties related with the noise (flesh contour and different gray levels surrounding the bone we want to segment). And, thanks to the MO methodology, we obtain the desired individual of Figure 5.18 (c) (minimizing the IOD_i/GD objective) without the experimental tuning of the weight parameters necessary in a single-objective minimization.

5.6 Discussion

As another improvement of the adapted GA that was proposed in this work, we designed a multiobjective methodology that used the SPEA2 algorithm, adapted to our application, using different evolutionary phases with different aims, and incorporating modifications in the density estimation technique to define the fitness of an individual. As a first approach, we used the characteristics of the proposed adapted GA, integrating them in the SPEA2 algorithm, especially regarding the



(a) (b) (c)

Figure 5.18: Results obtained in the segmentation of the humerus in medical CT images. (a) Example of the CT slices used. (b) Artificial reconstruction of the bone. (c) Best final result according the minimization of the IOD_i/GD objective.

production of new individuals using the classic operators and the new ad-hoc ones, and also taking the definition of different evolutionary phases with different aims. As a demonstration of the improvements, we compared the multiobjective approach with the classic GA for the same task.

In not very complex images the two approaches can obtain correct segmentations. The main problem of the GA approach is that it requires different parameter sets for the model in the two evolutionary phases defined. These parameters define the weights that the user considers for each of the objectives or energy terms, as the GA uses as fitness value of an individual a summed global energy that weights the different energy components. In the second GA evolutionary phase, these energy parameters must even be tuned for each image to obtain correct segmentations. That way, the process with the GA turns into an experimental tuning of the adequate weights for each image from multiple executions of the evolutionary processes.

In the multiobjective approach the previous problems are directly solved. Firstly, the aim of the two phases is obtained with the simple definition of the objectives that requires each phase. For example, the designer only needs to consider the IO and GD energy terms in the first phase to obtain the desired purpose of covering the objects to segment, which is a more natural decision. Secondly, the required experimental tuning of the weights in the GA case (or DE case) disappears, as the multiobjective optimization tries to optimize individually each one of the particular objectives. We also integrated two energy terms in the IOD_i/GD objective to join the characteristics of the adjustment to the object in this single objective. We also simplified the evolutionary process with the use of only one evolutionary phase in some segmentations.

The results with selected images to show the capabilities of the multiobjective

Table 5.5: Comparison between the single-objective methods (GA and DE) and the SPEA2 approaches.

	GA method	**DE method**	**SPEA2-GA approach**	**SPEA2-DE approach**
TAM Parameter tuning	Always	Always	Never	Never
Detection of irregularities (concavities, etc.)	Need specific parameters	Need specific parameters	Correct	Correct
Execution time	High	Medium	Very high	High

approach indicated that this approach outperforms the GA in several cases. For example, in objects with concavities or holes, the tuning required by the GA is critical to obtain correct results. The holes must be delimited by the distribution of the internal nodes, so the designer had to tune the weights of the objectives to obtain that delimitation aim (mainly the IOD weight). In the multiobjective approach, the designer can select a trade-off solution or nondominated individual form the Pareto Set depending on his main interest. For example, selecting those individuals that minimize the IOD objective to provide the best delimitations of the holes.

Additionally, we incorporated the DE approach in the SPEA2 algorithm, as an improved version of the multiobjective method, working as a genetic operator to generate the next population based on the individuals of the external file. This new version incorporates all the advantages of the previous MO approach, and also presents some new improvements. The integration of both methods allowed a minimization of the designer decisions because, as in the previous case of the basic SPEA2 algorithm, there is no need of the experimental tuning of the parameters or weights of the different energy components or individual objectives. Additionally, there is a simplification of the generation process of the next population, as only the DE operator is required, an advantage with respect to the use of a complete set of genetic operators with their tuned probabilities of application. Moreover, with respect to an evolutionary single-objective optimization, there is no need of different evolutionary phases focused on different segmentation aspects. Finally, the combination allowed a faster convergence of the image segmentations in the 2D and 3D domains.

The main drawback of the multiobjective approach was the complexity of the algorithm, compared with the single-objective optimizations of a GA or DE. As we mentioned, the calculation of S and R values in the SPEA2 algorithm (5.2.1),

used to establish the final fitness of an individual, is of complexity $O(M^2)$, where $M = N + \overline{N}$ (population size + archive size). With the inclusion of the density estimation technique in the final fitness, the final complexity increases to $O(M^2 log M)$ [110]. The adapted calculation of k-distances mitigates this last term, as we only calculate the distances among the nondominated individuals of the Pareto Front maintained in the external archive ($\leq \overline{N}$). Nevertheless, this drawback is clear and must be taken into consideration in the comparison with the use of a GA or DE for a single-objective optimization. Table 5.5 summarizes the main differences among the SPEA2 approaches and the single-objective ones (GA and DE methods).

Chapter 6

Practical application: Localization of the optic disc by means of evolutionary-optimized Topological Active Nets

6.1 Introduction

As a practical application, we used the evolutionary approaches in a specific domain, with the purpose to localize and extract the optic disc in retinal images. Firstly, we will explain the basic concepts and methodology of the use of retinal images considering the Genetic Algorithm (GA) developed, because in the timeline it was the first method used in this domain. After that, we will show some results with other evolutionary approaches useful for this type of images.

The retinal fundus photographs are widely used in the diagnosis of eye, cerebrovascular and other diseases. Automatically processing a large number of retinal images can help ophthalmologists to increase the efficiency in the medical environment. Figure 6.1 shows an example of a retinal image detailing the main characteristic structures, that is, the macula, the vessel tree and the optic disc. The optic disc is the entrance region of the blood vessels and also where the nerve axons enter and leave the eye. It is the brightest area in the image, it is a slightly oval disc and its detection is very important since it is used for blood vessel tracking and it works as a landmark to measure distances and identifying anatomical parts in the retina like the fovea.

The localization and segmentation of the optic disc have been previously per-

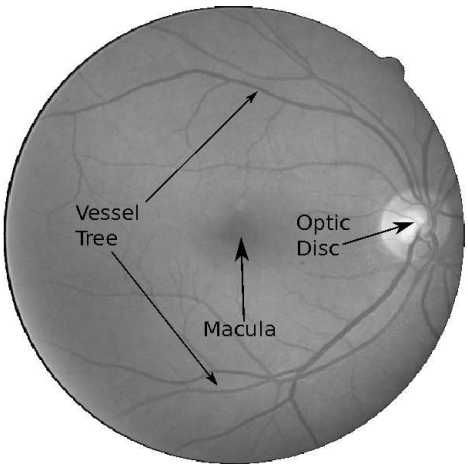


Figure 6.1: Example of retinal image focus on the macula including the main characteristic structures.

formed through several approaches, as we will explain in the next section. Some methods are focused on the localization of the optic disc center and others additionally segment the disc. Deformable models, concretely different snake models, have also been used in this application. The main problems are the noise presence in the areas in which the optic disc is located, the presence of blood vessels that cross the optic disc and that greatly difficult the segmentation, and the possible presence of lesions that could be associated with the optic disc.

In section 6.2 we detail the application of the TAN model to the localization of the optic disc. The modifications of the GA method (and, for extension, the rest of the evolutionary approaches) are depicted. The global search of the evolutionary approaches, together with new energy terms useful to the problem, allow the reduction of the commented problems such as the noise presence or the blood vessels internal to the disc. As we will explain, the methodology does not need pre-processing for the localization of the optic disc. Moreover, a segmentation of the boundary can be performed. In section 6.3 the results of the adapted evolutionary approaches are shown.

6.1.1 Previous work

There are several previous works on optic disc localization. Lalonde *et al.* [54] used pyramidal decomposition on the green channel to identify the potential regions containing the optic disc, and then extracted possible optic discs using Hausdorff based template matching, deleting the wrong results according to a confidence level. The

algorithm provided good results but depended on the completeness of the thresholded edge map. This task was accomplished by removing noisy edges and preserving the optic disc contour.

In the work of Li and Chutatape [57] the authors proposed a new method to automatically locate the optic disc. The candidate regions were first determined by clustering the brightest pixels in intensity image. For instance, if the number of pixels in a cluster was less than 100, the cluster was abandoned. Principal component analysis (PCA) was then applied to the remaining candidate regions. The minimum distance between the original retinal image and its projection onto disc space was located as the center of the optic disc.

There are other works that perform optic disc localization based on the use of vessel cues with proven robust methods. Niemeijer *et al.* [70] presented an automatic system to find the localization of the major anatomical structures: the optic disc, the macula, and the vascular arch. The structures were found by fitting a single point-distribution-model to the image which contains points on each structure. The method of Foracchia *et al.* [35] was based on a geometrical model of the direction of main retinal vessels (two parabolas with a common vertex). Hoover and Goldbaum [44] used a novel algorithm they called fuzzy convergence to determine the origination of the blood vessel network. It mainly tried to detect the blood vessels and then identify the convergence of all the vessels, that is considered the center of the optic disc. Their method uses brightness as a secondary feature for optic nerve detection. Using the results provided by ophthalmologists as the testing criterion, the authors reported a 100% of correct localizations in a set of 31 healthy retinas and 82% of correct detections in a set of 50 diseased retinas. The method of Tobin *et al.* [98] relied on the accurate segmentation of the vasculature of the retina followed by the determination of spatial features describing the density, average thickness, and average orientation of the vasculature in relation to the position of the optic nerve. These features were used to train and apply a Bayesian classifier which determined the likelihood of an image location being associated with the optic nerve.

Regarding segmentation, Chrástek *et al.* [21] used an automated method for the optic disc segmentation which consisted of 4 steps: Localization of the optic disc, nonlinear filtering, Canny edge detection and Hough transform. The nonlinear filtering was used as a method for noise reduction, which at the same time preserved the edges. Since the optic disc is a circular structure, the authors used the Hough transform as a method of circle detection. The transform gave them the center and radius of a circle approximating the border of the optic disc. The authors reported results with 97% successful localization and 82% successful segmentation.

The criterion of correctness was visual inspection in both cases. In the work of Abdel-Ghafar and Morris [1] the boundary of the optic disc was also estimated using a simple edge detector and the circular Hough transform. To simplify the detection, the blood vessels in the image were previously suppressed by morphological methods. A 24 radial vector set was defined to approximate the center of the optic disc as their origin, and then the image was resampled along these vectors to form a good representation.

The difficulty of the problem can be summarized in the work of Jelinek *et al.* [49]. The authors applied different steps for an integrated automated analyzer of the retinal blood vessels in the vicinity of the optic disc. First, they detected the optic disc using a combination of Butterworth filtering, Canny edge detection and morphological filters. The initial red plane image was first reduced in size using bilinear interpolation for efficiency. The image was then normalized using a high pass Butterworth filter. The local intensity standard deviation filter was then used to locate the optic disc as the region of greatest variation. In the next step the authors applied a greyscale morphological closing with a flat, disc-shaped structuring element to remove the edge of the blood vessels. Canny edge detection was then applied followed by a morphological closing with a disc structuring element to close any gaps in the perimeter of the optic disc. With this methodology the optic disc was well located in 13 of 20 images.

There are fewer works that have used active contours in the segmentation of the optic disc. Mendels *et al.* [64] used a two stage method. In a first stage, the image was processed using grey-level mathematical morphology to remove blood vessels regions, replacing them by pixels representative of the optic disc background. Then, a snake was manually placed around the optic disc and allowed to evolve onto its boundary. This snake [105] was an improved version including a Gradient Vector Flow (GVF) external energy force, which was calculated as a diffusion of the gradient vectors of a gray-level or a binary edge map derived from the image. The authors indicated that the accuracy of the method is highly sensitive to initialization together with the sensitivity of the snake to energy minima.

Osareh *et al.* [73] reported improvements on this previous work. They used a simple template matching approach to estimate the position of the disc center, which allowed the initialization of a snake automatically. First, they pre-processed the image through a closing operation (a dilation to remove the blood vessels followed by an erosion to restore the boundaries to their former position) to create a fairly constant region before applying a snake method. Then, a GVF-based snake was used to perform the segmentation. Moreover, the authors showed how the boundary

localization can be drastically improved using color mathematical morphology on the original color image.

Chanwimaluang and Fan [20] introduced methods for the automatic detection and extraction of blood vessels and the optic disc. The optic disc detection was performed with a two-step initialization for a snake active model. The authors used the local window based variance to select the initial center of the disc. Then they initialized the size and the number of contour points by estimating the local contrast and variance around the center. The authors also indicated that the initialization of size and shape of the snake model was critical to the final result. In the work of Lowell *et al.* [60] the optic disc localization was achieved using template matching (a specialized correlation filter) whereas the segmentation was performed with a snake model. According to the authors, no intervention was required as the algorithm automatically selected the general location of the center of the optic nerve head, and then fitted a contour to the optic nerve head rim. Comparing their results with the localizations obtained by ophthalmologists, they reported a 96% of correct localizations in a set of 100 fundus images.

As indicated by Xu *et al.* [107], deformable models can be roughly classified into free-form deformable models (snakes) and parametrically deformable models (ASM). The latter was used by Li and Chutatape [58], since they first used the previously mentioned PCA method to locate the optic disc and then a modified active shape model (ASM) to refine the optic disc boundary based on the point distribution model defined from the training sets.

In (Xu and Chutatape [106]; Xu *et al.* [107]) the authors used a deformable model which was modified and extended in two aspects. First, after each deformation, the contour points were classified into edge-point cluster or uncertain-point cluster by unsupervised learning, k-means algorithm. Second, the contour was updated by variable updating sample numbers. The clustering process self-extracted the uncertain contour points (typically points belonging to noise or vessels) from the correct edge points. The variable updating sample number combined global and local information of only the correct edge points to update the contour points after each radial deformation. The authors reported a better success rate of 94% on 100 testing images when compared to the results obtained by GVF-snake and modified ASM method. For the calculation of this rate, the groundtruth of the boundary was manually labeled under the supervision of ophthalmologists.

With a similar philosophy of detecting contour points, Carmona *et al.* [17] proposed a method which consisted of three stages. In the first phase, the eye fundus image was pre-processed. After that, they obtained a set of hypothesis points, which

were candidates to be in the optic disc boundary. Finally, the last stage consisted of selecting the most suitable points from the previous set and joining them properly with an ellipse to form the solution papillary contour. Inspired on this methods, Molina and Carmona [66] also proposed a new method for the detection of the optic disc contour. In this case, they used a Gaussian pyramid representation of the input image to obtain a subwindow centered at a point of the papillary area. In this subwindow they used a Laplacian pyramid to obtain a set of interest points (IPs) in two pyramid levels. Finally, a two-phase genetic algorithm is used in each pyramid level to find an ellipse containing the maximum number of IPs in an offset of its perimeter and, in this way, to achieve a progressive solution to the optic disc contour.

6.2 Modifications of the GA and the Topological Active Net

With our methodology, localization is based on the calculation of the center of the TAN, so a good segmentation is required. Additionally to the energy terms explained in previous chapters, we have incorporated new energy terms suitable for this application, as with the initial ones we obtained sub-segmentations. The reason was the high number of nodes that were located on the edges of the blood vessels. Therefore, the best final results only segmented part of the optic disc, and we added the next new energy components to solve these problems.

6.2.1 Circular structure

We know that the optic disc has a circular-like structure in any case. Therefore, a new component is added to give priority, in fitness terms, to the active nets of the genetic population with a circular structure. In order to do so we make use of the average radius $\overline{r}$, calculated as the average distance between the centroid of the whole active net and the external nodes. Then accumulate the sum of the differences between $\overline{r}$ and the distance from the net center to all the external nodes. The term is defined as:

$$CS[v(r,s)] = cs \sum | \ (|v(r,s) - \overline{v}|) - \overline{r}| \quad \text{for external nodes} \tag{6.1}$$

where $\overline{v}$ is the center of the mesh and $\overline{r}$ is the average radius to all external nodes. Therefore, nets with a circular structure will have less energy than others. The parameter *cs* weights the energy term.

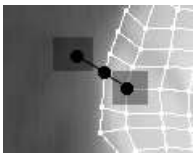


Figure 6.2: Example using the contrast of intensities. The energy of the central node is obtained using the difference between the mean intensity of the internal and external neighborhoods.

6.2.2 Contrast of intensities

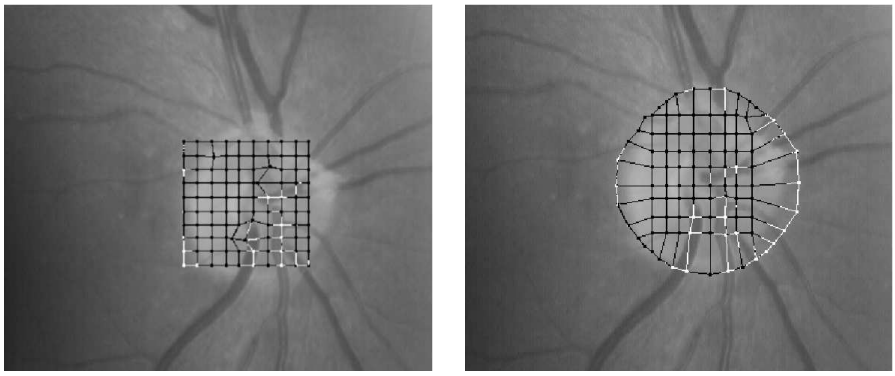
In retinal images the optic disc has brighter intensities than the area enclosing it. Ideally all the external nodes would be on optic disc edges and the internal ones on the bright intensities inside. Therefore, each external node would have a contrast of intensities between its outside and inside. With this new energy component, when the outside intensity is dark whereas the inside intensity is bright, the energy will be lower. We take these intensities after applying a mean filter to the image to avoid isolated erroneous intensities. Figure 6.2 shows an example of this. For an external node, we select its internal neighborhood and the correlative neighborhood outside the net. We calculate the mean intensity in a given neighborhood of 400 pixels (a square of 20×20) to minimize the effects of noise.

$$C[I(v(r,s))] = ci\left(\frac{\sum_{n \in \aleph(Ext[v(r,s)])} I[v(n))]}{|\aleph|} - \frac{\sum_{n \in \aleph(Int[v(r,s)])} I[v(n))]}{|\aleph|} \right) \qquad (6.2)$$

for external nodes.

where $Int[v(r,s)]$ is the internal neighborhood of the node $v(r,s)$, $Ext[v(r,s)]$ is the correlative in the outside, $\aleph$ is the chosen neighborhood, and ci is the corresponding weight parameter of this energy component. If the external nodes are correctly placed, the difference is negative.

The method is similar to the one used by Xu and Chutatape [106] with snakes. These authors used three terms as the external energy for the snake: the magnitude of the gradient, the difference of gradient orientation and the difference of intensity value. The magnitude of the gradient can efficiently provide the boundary location. However, as the authors comment, the point located on the edge of blood vessels also has high gradient magnitude. Hence, the gradient orientation information was introduced by the authors into the external energy.



(a) (b)

Figure 6.3: The two initialization processes. Best individual after several generations of the GA, in an evolutionary phase focused on correctly locating the TANs around the optic disc. (a) Square initialization. (b) Circular initialization.

6.2.3 Adaptations in the evolutionary process

Individual initialization

As we mentioned before, we know that the optic disc has a circular-like structure in all situations. The definition of the standard TAN uses initializations as squares. This means that the individuals of the initial population are TANs situated in random positions in the image, with random inter-distances among columns and rows of nodes and always with a squared shape. After that, these individuals have to be adapted to the optic disc with several deformations. In order to simplify these adjustments we initialize the population with all the external nodes uniformly distributed in a circular way. Therefore, we will obtain individuals with fewer necessary changes or deformations in order to reach the situation where the external nodes are well located, that is, near or over the optic disc contour.

In Figure 6.3 we show the two ways of initialization. The two nets shown correspond to the best individual in the population after few generations of the GA. In these generations the GA is focused on correctly locating the TANs (the explained $1^{st} phase$) of its population around the optic disc. At the end of this stage, the circular initialization provides better adjustments, which shows how right adjustments are performed very quickly with the circular initialization, simplifying the evolutionary process.

Evolutionary phases

If we run a simple genetic process we will not obtain good results. Once again, we need to use two different evolutionary phases as explained before, focusing on different energy components in each phase. The general idea is that the population can cover the optic disc in the first generations, whereas the next stage improves the adjustment to the optic disc contour progressively.

First phase The energy parameters allow the nodes to be outside the optic disc without a high penalization. One of the weight parameters of the energy components that facilitates this requirement is ω, which situates internal nodes in bright intensities and external nodes in dark pixels (Section 2.1.2). The other parameter is ci, which weights the new energy term "contrast of intensities" , which forbids external nodes to be situated on edges inside the optic disc. δ (Section 2.1.2) takes some importance to guide the external nodes to edges and also the roundness (cs) to preserve the circular-like structure. The other energy terms have less relevance. The shift operator is only applied in this evolutionary phase.

Second phase The parameter values are changed in order to search for a more homogeneous distribution of the internal nodes and adapt the TAN to the optic disc. Hence, exploration predominates in the first phase while exploitation dominates the second one. The application of the spread operator is decreased because it is not as necessary as in the first evolutionary phase. This phase finishes when it cannot obtain better results. This is determined when the fitness of the best individual in a given generation cannot beat any fitness of the best individuals in the previous 10 generations.

6.3 Results

In this section we present representative examples to show the capability of the methodology when dealing with retinal images with different levels of noise and different contrasts of intensities.

We used two different kinds of retinal images, one focusing on the optic disc, and another focusing on the macula. In all the examples the original image was used for the calculation of the external energy for both the external and the internal nodes. The images used in the next subsections were taken from the VARIA [102] and DRIVE [31] public datasets. The greyscale image resolutions are 768×584 (images from VARIA dataset) and 565×584 (images from DRIVE dataset). The

first sub-sections (6.3.1 to 6.3.5) explain the main characteristics of the localization results with our methodology, whereas sub-section 6.3.6 presents a test over all the images of those datasets. Finally, sub-section 6.3.7 shows different results obtained with the rest of the proposed evolutionary approaches.

In the first evolutionary phase, the values of the energy parameters can vary in a great range (and they are common to all the images), in order to obtain the desired result, in the sense that we want all the individuals of the genetic population covering the optic disc at the end of this phase. These used values were: $\alpha = 0.01$, $\beta = 0.0001$, $cs = 5.0$, $\omega = 2.0$, $\rho = 1.0$, $\xi = 0.0$, $\delta = 10.0$, $ci = 10.0$. In the second evolutionary phase the TAN parameters were experimentally set as the ones with which we obtained the best results. Table 6.1 shows the TAN parameters used in the different examples in the second evolutionary phase.

Regarding the GA, we used a tournament selection with a window size of 3% of a population between 800 and 1000 individuals and elitism of the best individual. The probabilities of the operators were also experimentally set, taking values in the ranges where the best test results were obtained. In first phase, these probabilities are: 0.5 for crossover, 0.0005 for mutation, 0.01 for spread, 0.001 for group mutation and 0.05 for the shift operator. In second phase, the probability of spread is reduced to 0.005 and the shift operator is not used because at this time these operators are less necessary. Finally, the number of generations of the first evolutionary phase was around 50. The second phase is finished when there is no improvement in the best individual (around 800 generations in most of the examples).

6.3.1 Justification of the circular structure energy term

We mentioned that the first new energy term introduced gives priority to the active nets with a circular structure. Without it we have notice that some genetic populations evolve losing individuals with this characteristic, even with the best adjustments. As we show in Figure 6.4, the best individual evolves losing a good adjustment because of the lack of this energy term. Finally, the evolutionary process gives a sub-optimum segmentation, so the location of the center of the optic disc is not as good as we desire. The external nodes tend to be located over the main blood vessels, because these positions provide a local minimum as the energy terms related with the gradient image are minimized (Eq. 2.7).

6.3.2 Justification of the energy term of the intensity contrast

The second energy component introduced is the “intensity contrast”. The minimization of this term keeps individuals with external nodes on the optic disc edges,

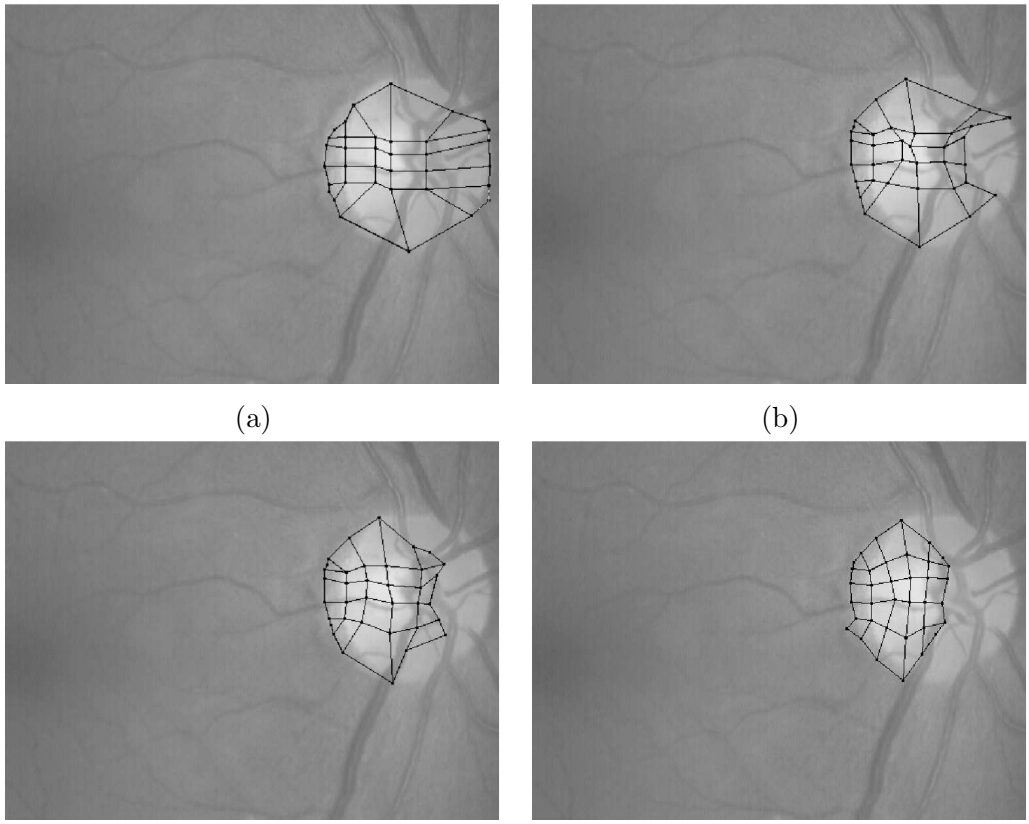


Figure 6.4: Example of GA evolution without the circular structure energy term. Best individual in different generations of the evolutionary process. (a) Best individual at the end of the first phase, generation 50. (b)(c) Intermediate generations. (d) Final result, generation 5000.

Table 6.1: TAN parameter sets in the segmentation processes of the examples (2^{nd} evolutionary phase).

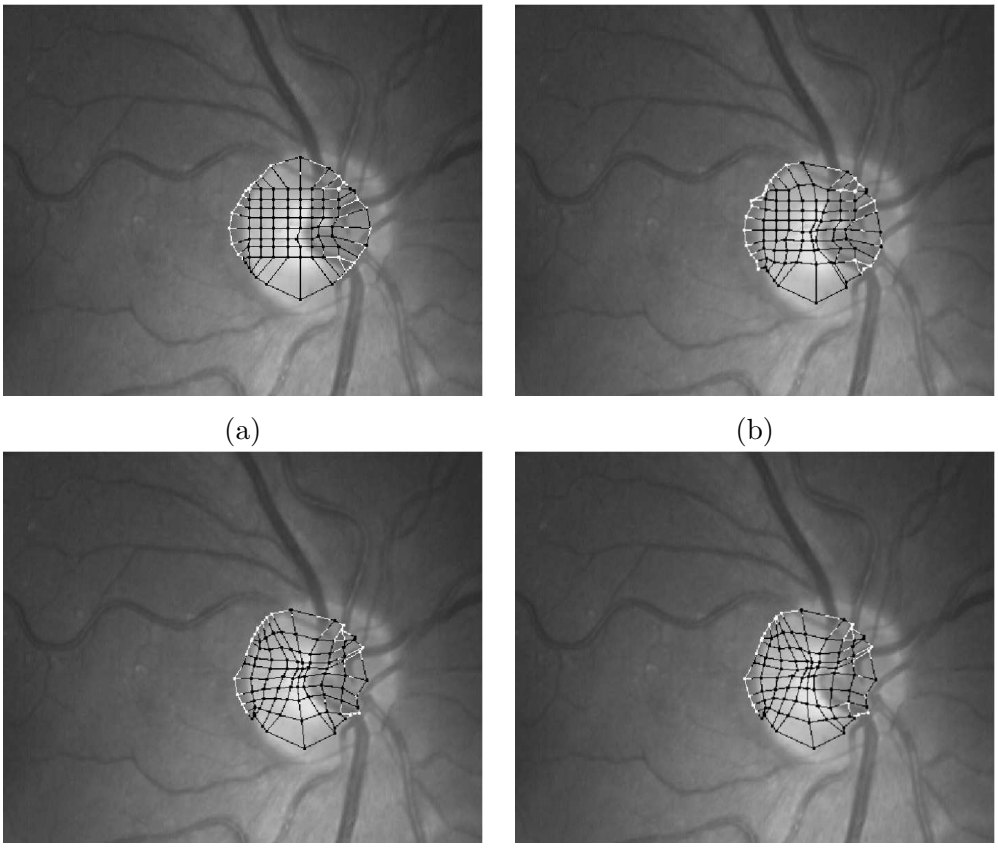
Figure	*Size*	α	β	cs	ω	ρ	ξ	δ	ci
6.7	10×10	3.0	1.5	100.0	10.0	3.0	4.0	5.0	15.0
6.8(a)	8×8	3.0	1.5	200.0	10.0	4.0	4.0	7.0	15.0
6.8(b)	10×10	3.0	1.5	200.0	10.0	4.0	4.0	7.0	15.0
6.8(c)	15×15	3.0	1.5	200.0	10.0	4.0	4.0	7.0	15.0
6.8(d)	20×20	3.0	1.5	200.0	10.0	4.0	4.0	7.0	15.0
6.9	7×7	1.0	0.5	100.0	10.0	4.0	4.0	7.0	10.0
6.10(a)	10×10	3.0	1.5	10.0	10.0	3.0	4.0	7.0	15.0
6.10(b)	10×10	1.0	0.5	100.0	10.0	4.0	4.0	7.0	15.0
6.10(c)	10×10	3.0	1.5	100.0	10.0	4.0	4.0	7.0	15.0
6.10(d)	10×10	1.0	0.5	100.0	10.0	4.0	4.0	7.0	15.0
6.12	10×10	2.0	1.0	200.0	10.0	3.0	4.0	6.0	15.0
6.14(d)	10×10	2.0	1.0	100.0	10.0	4.0	5.0	7.0	10.0
6.15(d)	10×10	2.0	1.0	100.0	10.0	4.0	5.0	7.0	10.0
6.16(d)	10×10	2.0	1.0	100.0	10.0	4.0	5.0	7.0	10.0

by means of checking a contrast of intensities between the inside and outside of the external nodes. Without this energy term, the individuals or TANs of the genetic population tend to locate their nodes on internal blood vessels obtaining wrong boundaries. Figure 6.5 shows an example, where the external nodes are not correctly located over the surface of the optic disc. The final TANs tend to be as contracted as possible, placing the nodes on blood vessels, although maintaining a circular structure because of the circular energy term.

6.3.3 Images focused on the optic disc

The first kind of images we worked with has the optic disc region in their center. In these images the optic disc section is easy to obtain because this region is bigger than in other images we will analyze and the contrast of intensities is higher.

First, we tested a greedy process to test its performance on these images. The adjustment process consists of minimizing the energy functions following the idea explained in Section 2.3. Summarizing, the initial mesh is placed over the whole image and, in each step, the energy of each node is computed in its current position and in its nearest neighborhood. The position with the lowest energy value is selected



(a) (b)

(c) (d)

Figure 6.5: Example of GA evolution without the energy term of the intensity contrast. Best individual in different generations of the evolutionary process. (a) Best individual at the end of the first phase (generation 50). (b)(c) Intermediate generations. (d) Final segmentation, generation 5000.

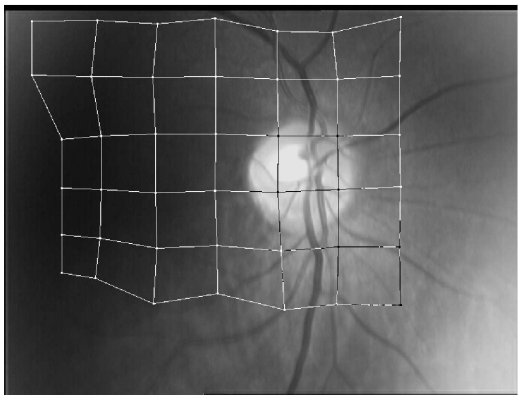


Figure 6.6: Final result obtained with the greedy segmentation process. The optic disc boundary was not detected.

as the new position of the node. The algorithm stops when no node in the mesh can move to a position with lower energy.

As we thought, the results obtained are unacceptable. Figure 6.6 shows the final result obtained from an example using the greedy process. The greedy algorithm easily falls in a local minimum because of the noise in the image, so in a few iterations the adjustment cannot continue. The optic disc limits are not found so we obtain wrong detections.

In the following examples we show 4 images in each row that implies one execution per row (Fig. 6.7 and Fig. 6.9): The first two images are localizations with the original image, and the last two images are a zoom in the optic disc area, so we can see the final adjustment in better detail. The first evolutionary phase is very short, and the results obtained at the end of this phase, or even before, let us adjust a circumference to the optic disc with high precision. The circumference is the best adjusted one to the external nodes, and it is depicted to show the quality of the localization of the optic disc with the optimized TAN. In the final segmentations, we can see how practically all external nodes are correctly located on the optic disc contour.

Regarding representative computation times, the execution time of the first phase with 10 generations, with around 800 individuals, a 10×10 TAN, is usually between 20 and 30 seconds, in an Intel Core 2 2.4 GHz. Nevertheless, the process can provide a TAN well located to adjust a good circumference in 3 or 5 generations, being even faster. The processing time of the GA process in one generation (in both phases) basically depends on the size of the net and the population. The image size is not relevant.

We present two examples with different difficulty. The first one in Figure 6.7 (top row) shows an input image where the results were perfect, in terms of localization. This implies the detection of the center of the optic disc and its comparison with a center provided by ophthalmologists, as we explain in Section 6.3.6. The second one in the next row corresponds to a noisy image and an optic disc with fuzzy edges. However, the genetic process, even in this situation, can obtain good localizations and the adjusted circumference encloses correctly the optic disc boundaries.

The two bottom rows in Figure 6.7 correspond to the same images but this time using a circular initialization in the initial TANs of the genetic population. We can notice the improvement in quality using the circular initialization, especially with difficult localizations.

Use of internal nodes to detect blood vessels

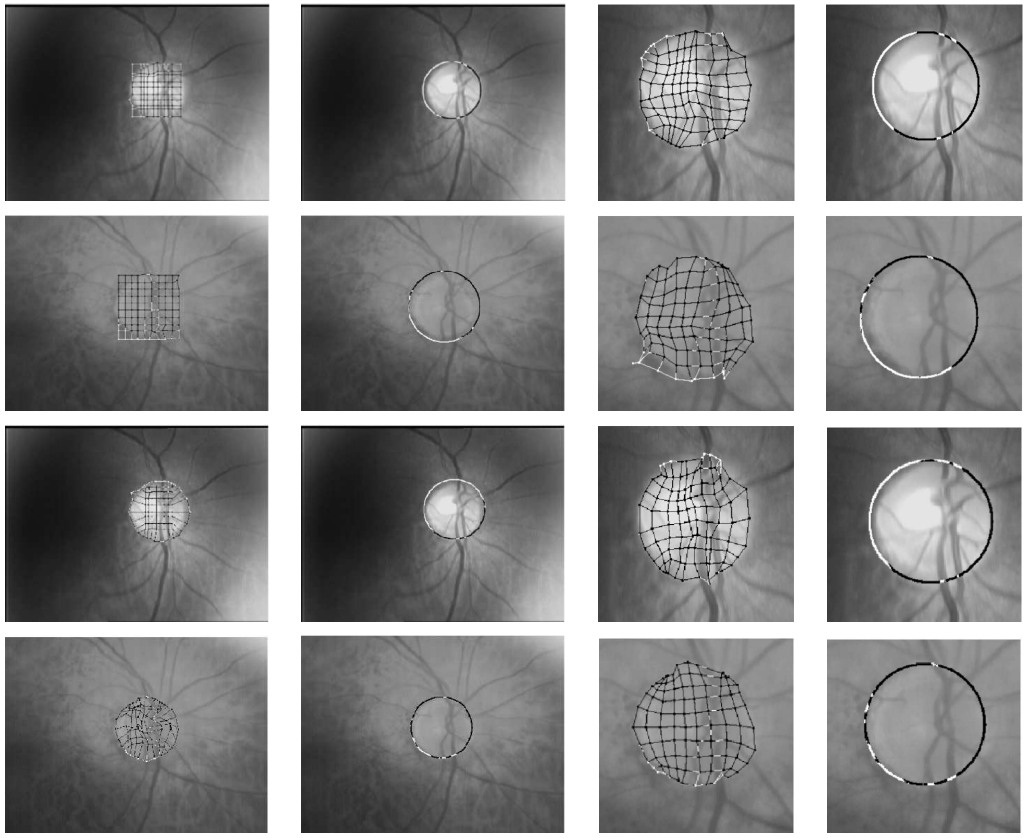
Even though the detection of blood vessels is not the objective of the localization, the information provided by the internal nodes can be used for this task.

One of the features of a TAN is the modeling of the internal topology of the segmented object that can be done with the internal nodes. Inside the optic disc, blood vessels and background have different tonalities in intensity range. These blood vessels are the candidates to be detected by the internal nodes.

The external energy term will benefit the location of internal nodes in bright areas avoiding the dark ones. With low resolutions of nodes (TAN size) the internal nodes do not give useful information, because the energy term that searches for more homogeneous distributions is more important. Nevertheless, when we increase the size of the TAN, the internal distribution can give information about the internal topology.

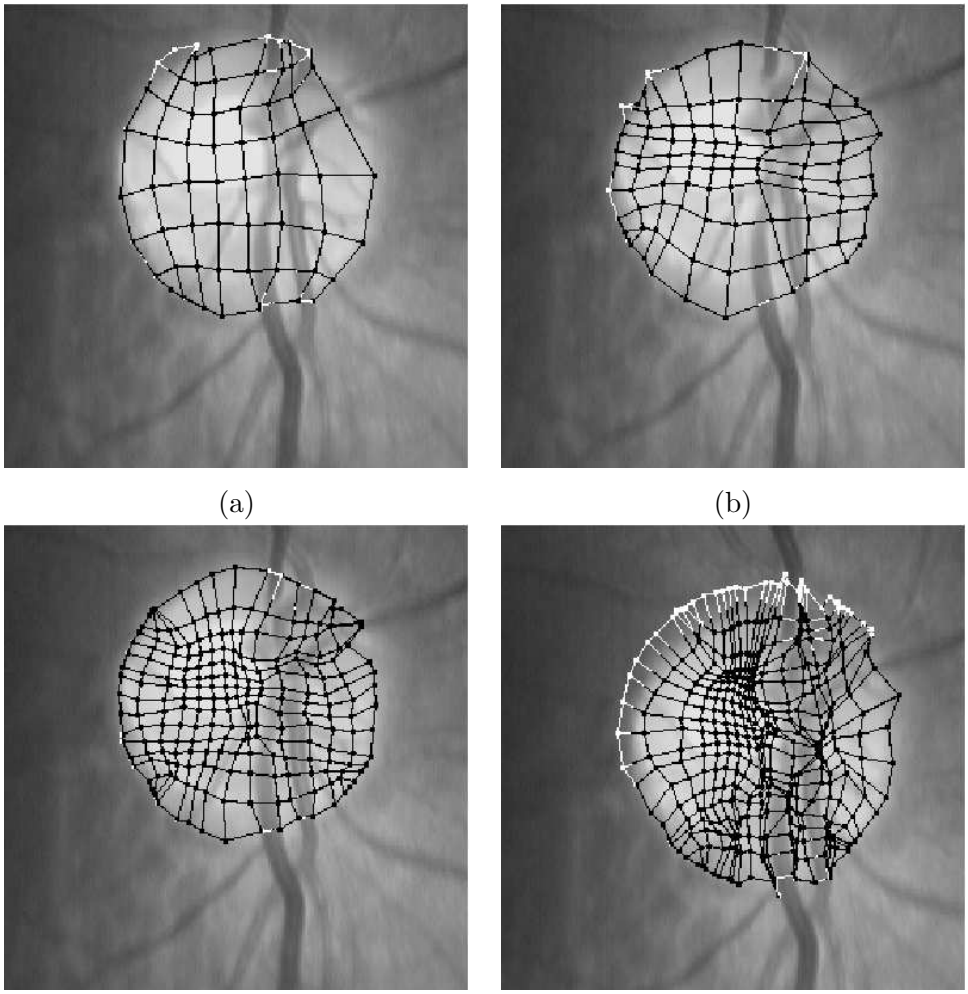
In Figure 6.8 we can see the results obtained with several TAN resolutions. As we said before, the internal nodes tend to avoid the blood vessels. This is clearer in the final image, with a TAN size of 20×20. Therefore, with high TAN resolutions we would be able to identify the regions were vessels are located.

One possible way of detecting blood vessels from the mesh information could be an analysis of tension in the connections between the internal nodes. That is, if there is a high tension, it could be because these nodes escaped from pixels on vessels, and they could be placed near their boundaries. If the intensity in the pixels where the nodes are situated is clearly brighter than in the pixels where their connection crosses, that could indicate that exists a blood vessel flowing through that region. In this situation, we could detect vessels and their width.



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

Figure 6.7: Segmentation results, used for localization, on images with the optic disc in the center. The first two upper rows use the square initialization and the two bottom rows use the circular initialization. (a) Best individual in generation number 10 (change of evolutionary phase). (b) Optic disc segmentation adjusting a circumference to the external nodes in (a). (c) Final segmentation. (d) Optic disc segmentation adjusting a circumference to the external nodes in (c).



(a) (b)

(c) (d)

Figure 6.8: Results using different TAN resolutions. (a) 8×8. (b) 10×10. (c) 15×15. (d) 20×20.

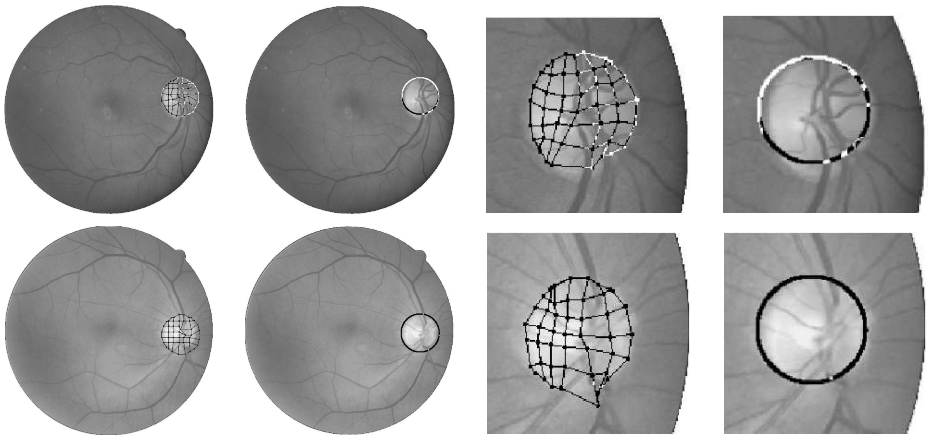


Figure 6.9: Segmentation results, used for localization, over images centered on the macula. (a) Best individual in generation number 10. (b) Optic disc segmentation adjusting a circumference to the external nodes in (a). (c) Final segmentation. (d) Optic disc segmentation adjusting a circumference to the external nodes in (c).

6.3.4 Images focused on the macula

The other kind of images we analyzed are centered on the macula, so the optic disc is located on a side. The localizations are more difficult because the contrast of intensities is lower with respect to the images centered on the optic disc, and also that in this case the image of the optic disc is smaller. Moreover, the region where the optic disc is placed is near the borders of the retina domain, so it is more difficult to obtain individuals with the external nodes surrounding the optic disc boundary. In this situation, the initial population would be composed with less individuals that cover approximately the optic disc contour. Nevertheless, the GA obtains almost perfect results. Two examples with images from the DRIVE database [31] are shown in Figure 6.9, using the circular initialization. The correct detection on the second image shows the robustness of the GA, as this image has a very poor level of contrast of the optic disc.

6.3.5 Retinal images with exudates and other pathologies

The worst situations to detect the optic disc are retinal images with certain illnesses and exudates that show several bright areas. These areas could be confused with the optic disc. We tested our evolutionary methodology to prove its efficiency in these special and difficult situations. The images used were provided by the Ophthal-

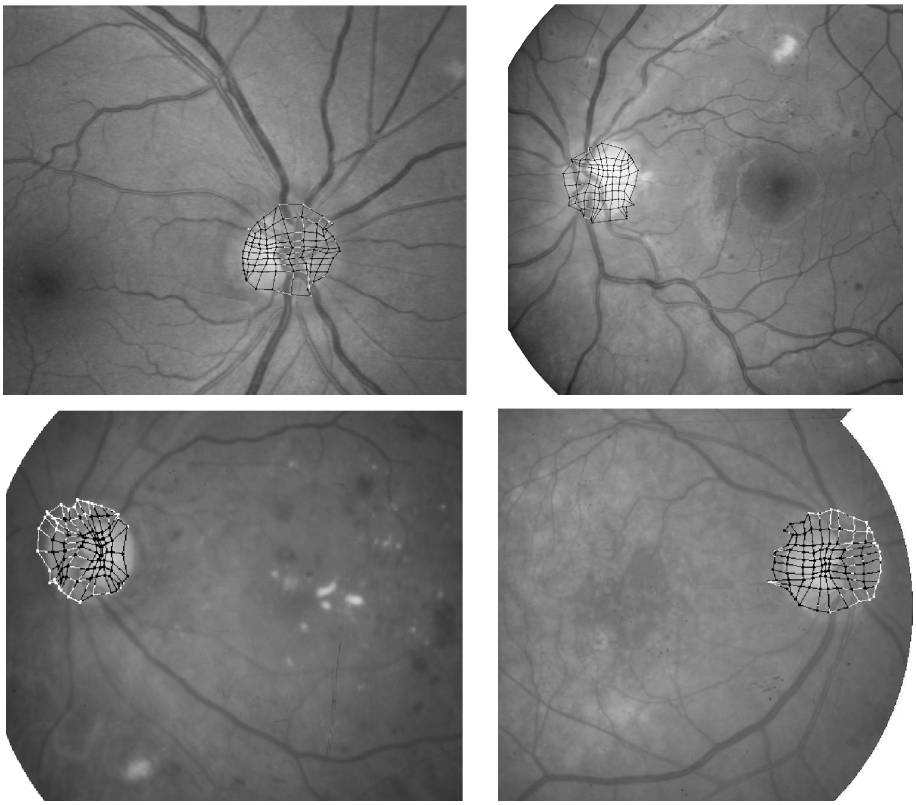


Figure 6.10: Localization examples in images with exudates and other bright areas.

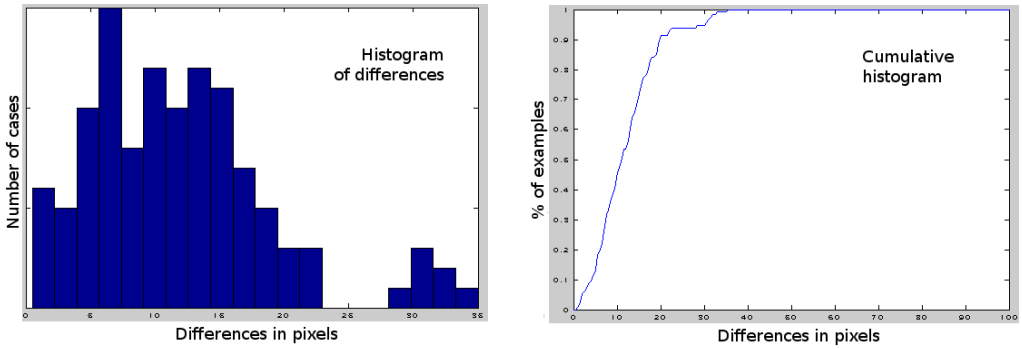
mologist Technological Institute of Santiago de Compostela (Spain) [46]. In Figure 6.10 we can see the good behavior, with correct detections in all cases. The GA escapes the local minima that would correspond to other bright areas, and obtains acceptable detections over the optic disc, even with high noise and the fuzzy edges of the optic disc.

6.3.6 Test on public databases of retinal images

To test qualitatively our method we have performed an analysis using public databases of retinal images.

Test on VARIA public database

The first database used for a complete test was VARIA [102], which contains 233 images of size 768×584. These images belong to 139 different individuals, they are centered on the optic disc and they do not have relevant pathologies. The images were taken with a Topcon TRC-NW100 non-mydriatic retinal camera.



(a) (b)

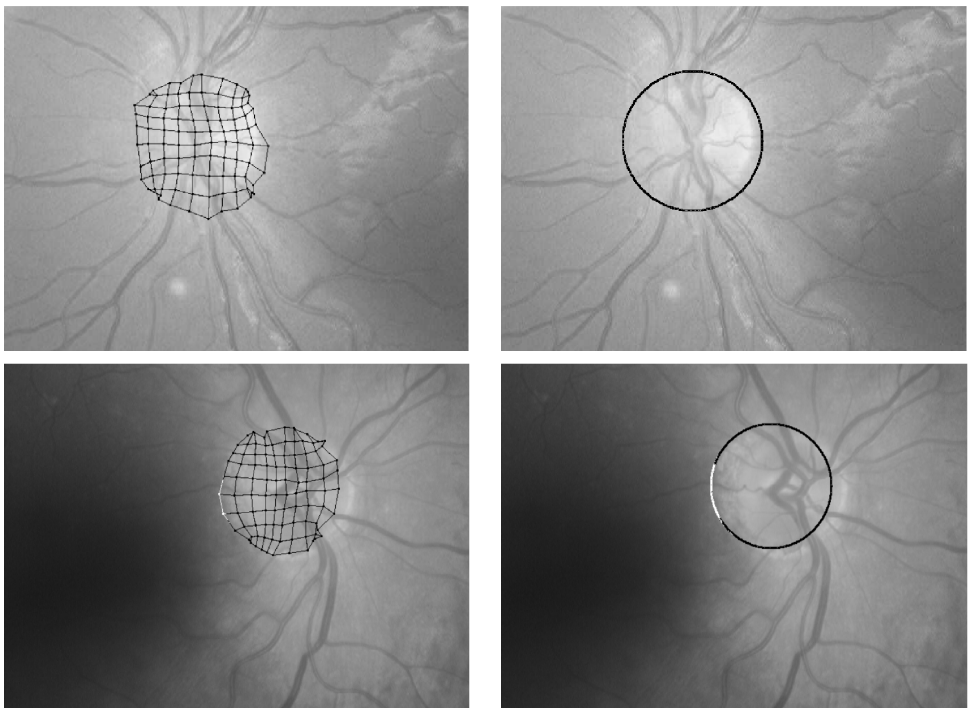
Figure 6.11: Analysis of differences among ophthalmologist localizations and results of the method. (a) Histogram of all differences. (b) Cumulative histogram.

We have performed an analysis of the localization success rate. To determine a correct localization of the optic disc we have followed the same criterion of Hoover and Goldbaum [44]. Using this criterion, the optic disc detection is successful if the computed point is within a 60 pixels radius from the ground truth location provided by an ophthalmologist. Our hypothesis identifies the optic disc center as the center of the net. Thus, the localization success rate was 100%.

Two ophthalmologists from the Ophthalmologist Institute of Santiago de Compostela (Spain) have identified the optic disc center in each image. The mean of both ophthalmologists is compared to the results of our method. In Figure 6.11(a), we present the histogram of all the differences and, in Figure 6.11(b), the cumulative histogram. The OD diameter in these images usually takes a value between 100 and 220 pixels, and the maximum difference obtained was under 35 pixels (all cases inside the OD).

In Figure 6.12 two of the most difficult examples for the localization are shown. The TAN parameters used in all localizations were set to the same values, trying to make the process parameter independent (Table 6.1). In these images we can see some difficulties in the localization. In the first image (upper row), the detection has problems due to the irregular illumination of the retinal image. The external nodes of the final TAN do not have a perfect circular shape due to the fuzzy contour of the disc. Nevertheless, the detection of the optic disc is perfect.

The second image in the bottom row is another difficult example with a fuzzy optic disc contour. The localization is again correct, despite the irregular illumination in the image. Since we have used the same parameters in all the selected images of the database, these examples show a high independence of the methodology with



(a) (b)

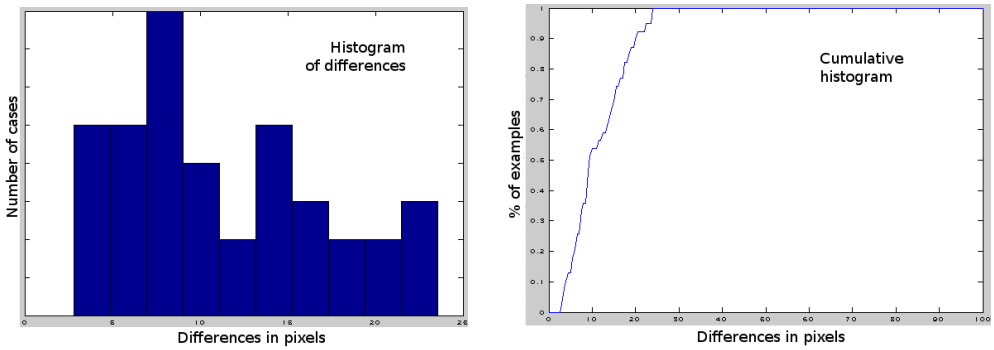
Figure 6.12: Two localization examples with difficult images from the VARIA database. (a) Final segmentations. (b) Optic disc detection adjusting a circumference to the external nodes in (a).

respect to the used TAN parameters.

Test on DRIVE public database

We also tested our method on the DRIVE image set [31] in the same conditions as explained on the previous dataset. The DRIVE images have been randomly selected from a diabetic retinopathy screening program in the Netherlands. This program consisted of 400 diabetic subjects from 25 to 90 years old. The DRIVE database has 40 images, 33 of them do not show any sign of disease whereas 7 show signs of mild early diabetic retinopathy. The resolution of the images is 565×584 and are centered on the macula.

In this case, the localization success rate obtained was again 100% with the same methodology and with the results provided by the ophthalmologists. The TAN parameters in all the images were set to the same values of the previous database. Figure 6.13(a) shows the histogram of all differences meanwhile Figure 6.13(b) presents the cumulative histogram. Using this dataset, the OD diameter usually takes a



(a) (b)

Figure 6.13: Analysis of differences among ophthalmologist localizations and results of the method. (a) Histogram of all differences. (b) Cumulative histogram.

value between 70 and 130 pixels and the maximum difference was under 25 pixels (all cases inside the OD).

6.3.7 Segmentation of the optic disc using multiobjective and Differential Evolution approaches

All the optic disc detection and segmentation were developed and tested using the proposed GA. However, we also tested the methodology using the other evolutionary approaches proposed in the work.

The application of Differential Evolution (DE) is direct, under the same conditions we explained in this chapter. We just substituted all the mechanisms of producing individuals as explained in Chapter 4. An example of this application was previously shown in Figure 4.8 (d). As we explained, with the use of DE, we had less evolutionary parameters to tune and faster segmentations with respect to a GA.

We also tested the segmentation with the proposed multiobjective approaches. In this case, we can define the objectives we desire to obtain correct results. The first example of segmentation of Figure 6.14 corresponds to a retinal image with the optic disc in the center. This is a typical retinal image where the region of the optic disc is easy to detect. In this segmentation, we used the multiobjective version using the classic GA and two different phases with the objectives defined in sub-section 5.2.3, that is, Gradient Distance GD and In-Out (IO) objectives in the 1^{st} phase, and GD, Distance In-Out (IOD) and internal energy objectives in the 2^{nd} phase. In Figures 6.14(b) and 6.14(c) the best individuals taken from the Pareto Front that minimizes the GD and IOD objectives are shown. Finally, in Figure 6.14(d)

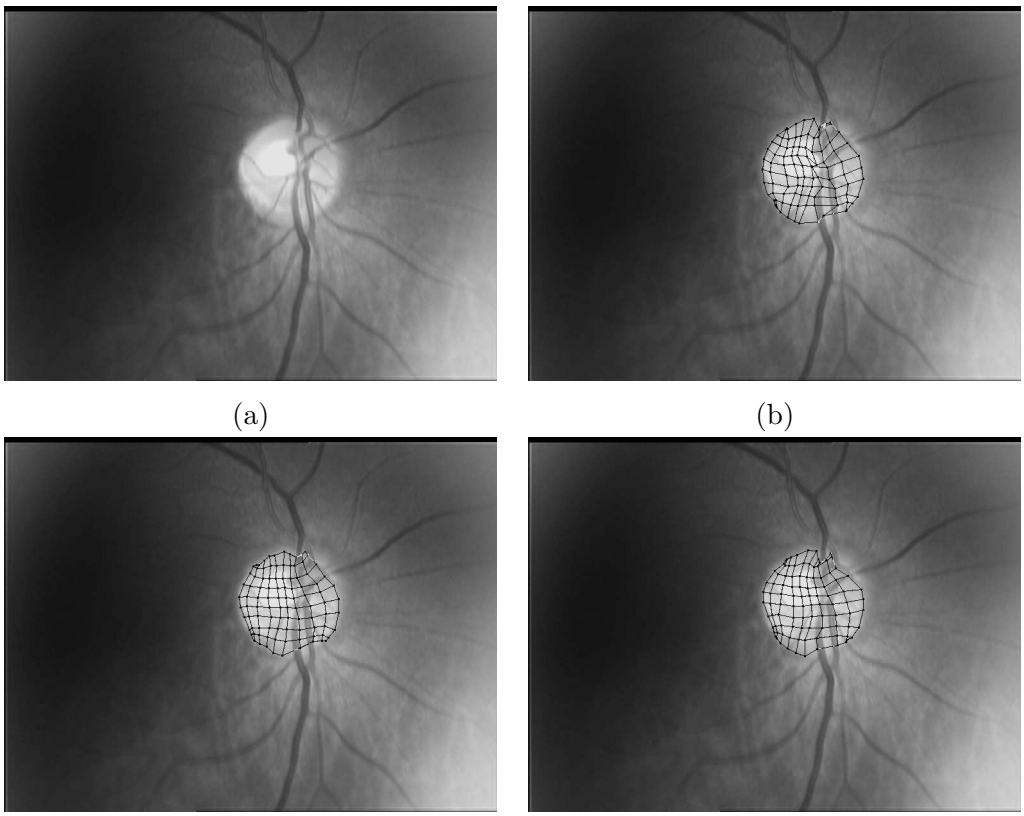


Figure 6.14: Results obtained in the segmentation of a typical retinal image using multiobjective optimization. (a) Image to segment. (b) Individual from the Pareto Front that minimizes the GD energy term. (c) Individual from the Pareto Front that minimizes the IOD energy term. (d) Best nondominated individual according to a particular parameter set.

we extracted the individual that performed the best adjustment according to the parameters detailed in Table 6.1. These parameters were chosen because they were used with the GA methodology.

In Figure 6.15 we needed to introduce the ad-hoc energy terms as new domain-specific objectives because with only the three previous objectives the multiobjective process had problems to detect the boundary of the optic disc. Figure 6.15(b) minimizes the GD objective because situates some external nodes on the blood vessels that provides a high contrast of intensity. Nevertheless, without the new added energy terms the problem would be worse because more external nodes would fall in the main vessel that crosses the optic disc. This effect had not occurred in the previous example because it had a better contrast of intensities in the optic disc.

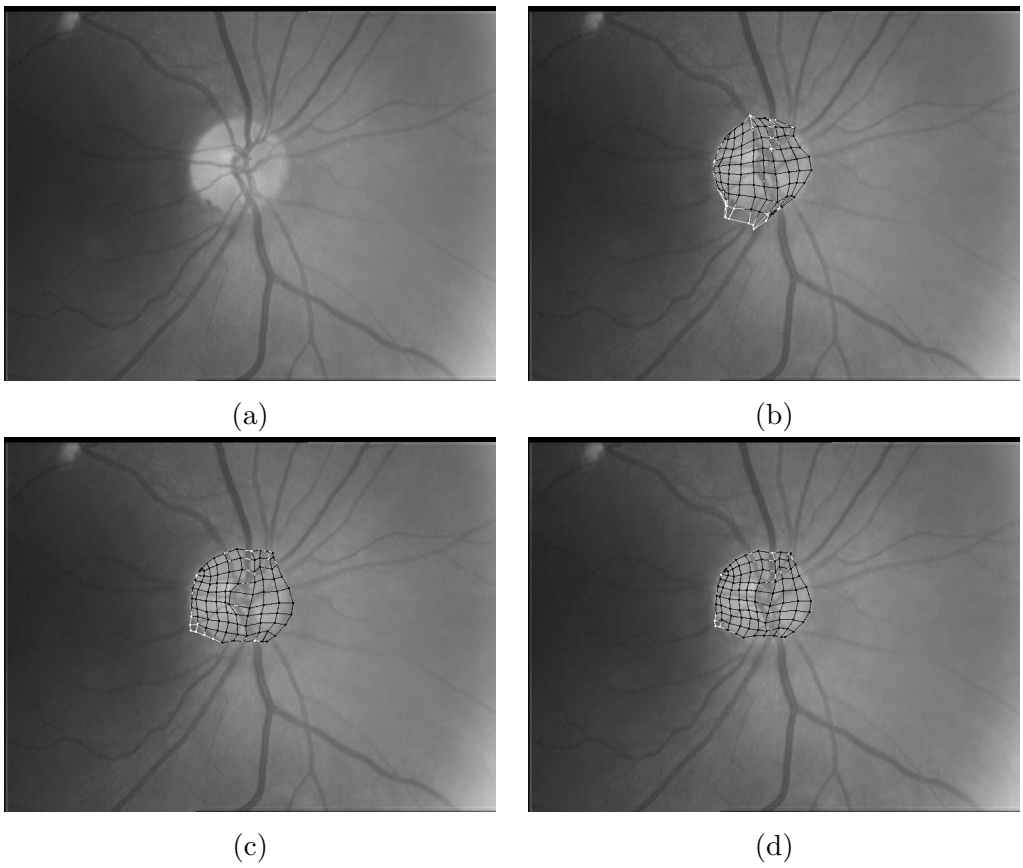


Figure 6.15: Results obtained in the segmentation with the incorporation of the ad-hoc energy terms as objectives. (a) Image to segment. (b) Individual from the Pareto Front that minimizes the GD energy term. (c) Individual from the Pareto Front that minimizes the IOD energy term. (d) Best nondominated individual according to a particular parameter set.

Figure 6.15(c) situates more internal nodes on the brightest area so it minimizes the IOD objective. Figure 6.15(d) represents the best individual of the Pareto Set that minimizes a given energy parameter set using, for example, the values $ci = 100.0$ and $cs = 10.0$ to weight these new objectives.

In the final example of Figure 6.16 we tested the methodology with a retinal image with special difficulties as it presents pathologies. The image has several bright areas corresponding to pathologies in the retina. These areas can easily confuse the segmentation process because of their bright values and contrast of intensities as the optic disc has. The vascular tree that crosses the optic disc also provides a gradient as the external contour, which can easily induce the fall in local minima, as in the previous case. In this more complicated example we need again to introduce the

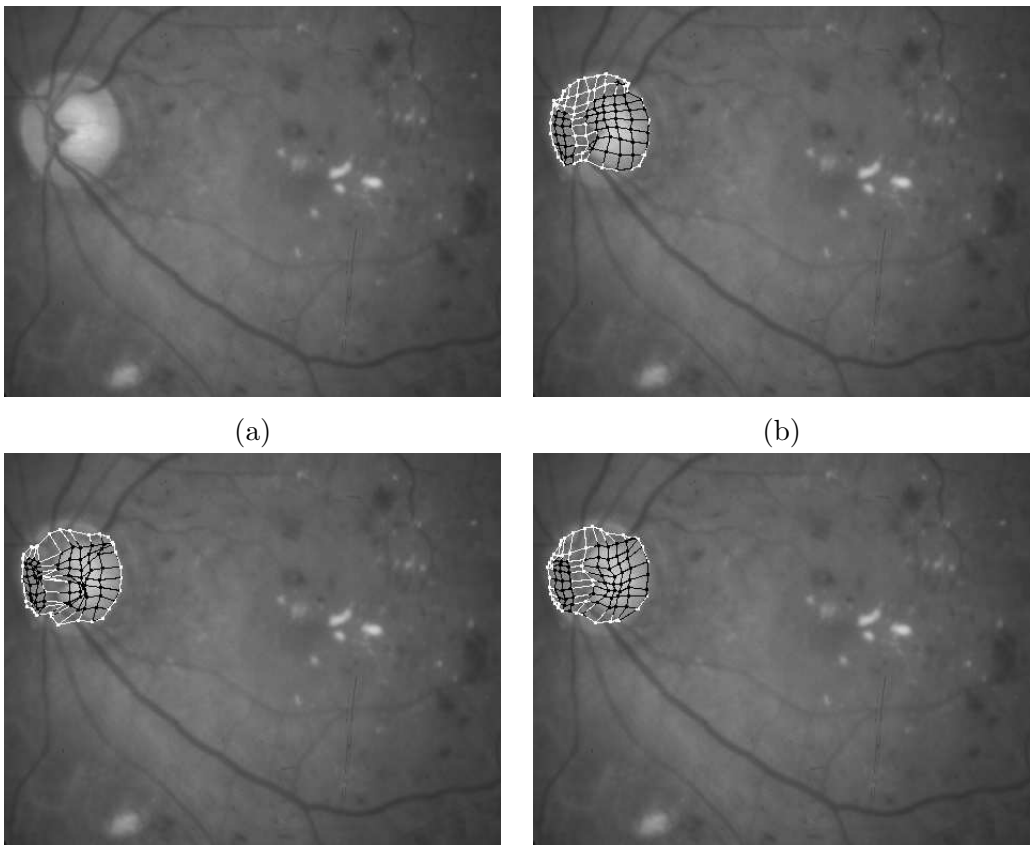
Circularity and the Contrast of Intensities energy terms as two new objectives in the second phase. These two added energy terms help to escape those possible local minima.

In Figure 6.16, once again, the best individual of the Pareto Front that minimizes the Gradient Distance and Distance In-Out energy terms are shown (Figures 6.16(b) and 6.16(c)). The TAN of Figure 6.16(d)) corresponds to the individual from the Pareto Front that minimized a summed energy according to the parameter set shown in Table 6.1. The weights of the new energy components are the same as the previous example ($ci = 100.0$, $cs = 10.0$).

Hence, Figure 6.16(b) has the best adjustment to the fuzzy external contour of the optic disc. Figure 6.16(c) minimizes the IOD objective as it moves the internal nodes to the brighter areas of the optic disc. Finally, Figure 6.16(d) performs an acceptable trade-off segmentation, where additionally the internal node distribution indicates the location of the vascular tree crossing the disc.

Finally, we also applied the hybridized SPEA2-DE method. We also tested the hybridized proposed method with the aim of localizing and extracting the region of the optic disc following the same idea as the other approaches. In this case, once again, we used two different phases (100 generations in the first phase and 5000 generations in the second phase) to obtain better results: a first one with the objectives IOD_i/GD and β, to localize the optic disc, and a second evolutionary phase adding a 3^{rd} objective (Circularity). In this version we added the ad-hoc energy term Circularity (to potentiate individuals in the population with circular shapes). In Figure 6.17 an example of segmentation using a retinal image taken from the Varia [102] dataset is shown. In the Figure, some representative individuals were taken from the segmentation process. Figures 6.17(a),(b) and (c) show the best individuals from the final Pareto Front that minimize the objectives IOD_i/GD, Circularity and β, presenting a correct adjustment to the edges, a quasi-perfect circular shape and a high level of smoothness, respectively. Note that in Figure 6.17 (a) the TAN minimizes the IOD_i/GD objective as all the external nodes are in the fuzzy boundary and the internal nodes tend to move to optic disc intensities, escaping from the internal blood vessels.

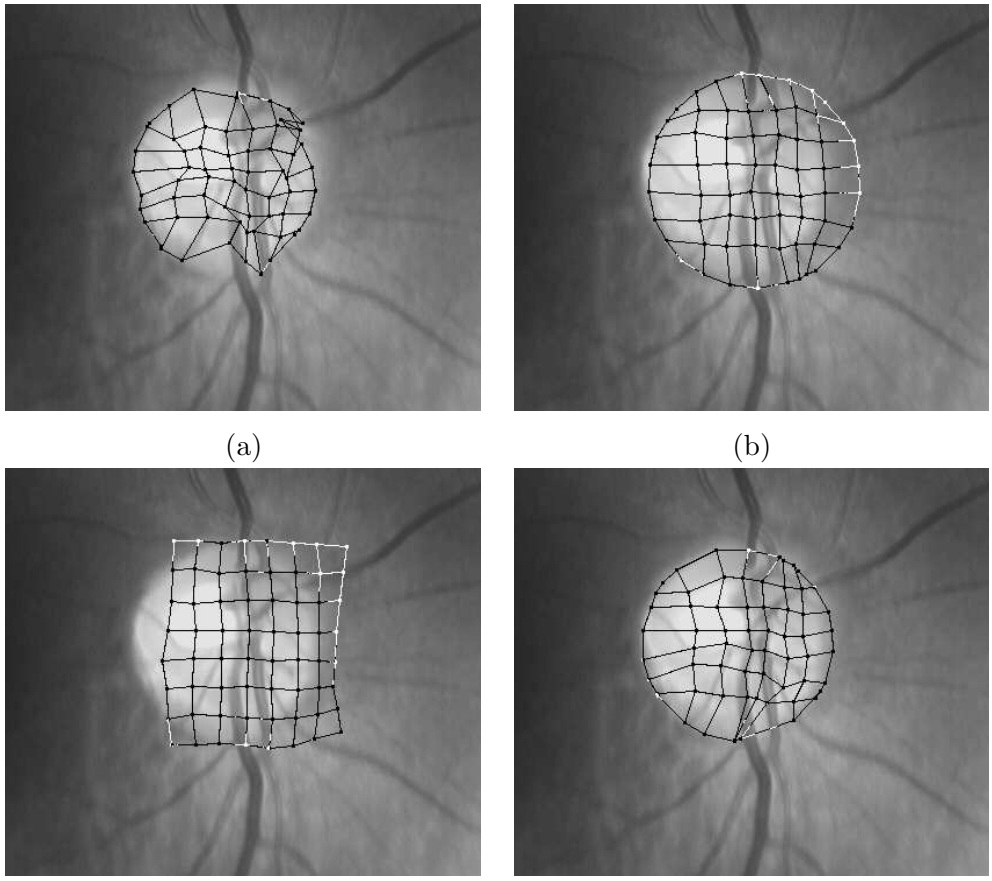
Finally, an individual was taken from the middle of the final Pareto Front in Figure 6.17 (d). This individual presents an adequate balance among the different characteristics of the objectives, performing a correct segmentation of the optic disc.



(a) (b)

(c) (d)

Figure 6.16: Results obtained in the segmentation of a retinal image that presents pathologies. (a) Image to segment. (b) Individual from the Pareto Front that minimizes the GD energy term. (c) Individual from the Pareto Front that minimizes the IOD energy term. (d) Best nondominated individual according to a particular parameter set.



(a) (b)

(c) (d)

Figure 6.17: Results obtained in the segmentation of the optic disc in retinal images. (a) Best final result according the minimization of the IOD_i/GD objective. (b) Best final result according the minimization of the Circularity objective. (c) Best final result according the minimization of the β objective. (d) Final active model taken from the middle of the Pareto Front.

6.4 Discussion

Xu *et al.* [107] stated that "the methods of optic disk segmentation can be separated into two steps: optic disk localization and disk boundary detection". As a practical application of the evolutionary segmentation methods, we developed an application of the TAN model which integrates the two steps, focusing on the localization one, but obtaining the segmentation of the optic disc as well. We have applied the different evolutionary approaches for the optimization of the TANs. Some domain information has been also incorporated in the TAN model, and the evolutionary approaches mainly needed two evolutionary phases in order to obtain correct final results.

Previous works that have used snakes for the segmentation required pre-processing, typically by the application of different filters and by removing the blood vessels, to minimize the incorrect boundary detection, being essential steps for accurate convergence. Opposite to this, the global search of the evolutionary methods provided a high level of robustness in all situations, so we were able to use the original images without pre-processing. In addition, the deformable model provided the optic disc contour and, moreover, the internal nodes could give us information about its internal structure as well as the position of the blood vessels across the disc.

The proposed methods for the optimization of the TAN model were tested with several images and with two different kinds of retinal images. In the different examples the new approach achieved a correct localization of the optic disc. Without a detailed statistical analysis the results indicate that the genetic algorithm shows a low sensitivity to noise or to the presence of blood vessels and it obtains remarkable correct detections in images with fuzzy or with poor contrast in the optic disc. This makes the methodology applicable to a high number of images and visual scenarios, even on images with bright areas corresponding to different pathologies. We performed the validation with the public datasets using only the GA method for simplicity, but we also showed representative results using the other proposed evolutionary methods.

The methodologies provided a robust behavior, although the single-objective GA and DE approaches needed a tuning of the parameter set which weights the different energy terms. However, as explained, this drawback is overcome with the multiobjective approaches. Thus, we can experiment with different objectives to obtain the desirable Pareto Front, according to the characteristics of the objectives that were selected.

Chapter 7

Conclusions

The Topological Active Net (TAN) and its 3D extension, the Topological Active Volume (TAV), are discrete implementations of an elastic $n-$dimensional mesh with interrelated nodes that integrates features of region–based and boundary–based segmentation techniques, generally called Topological Active Model (TAM). Their adjustment was previously performed by a greedy local search, segmentation technique that presented several limitations.

In this thesis we designed and implemented new approaches for the optimization of the Topological Active Models. All the methods used different evolutionary techniques that overcome several drawbacks of the initial method defined. The approaches included the characteristics of the evolutionary methodologies and also adapted some important characteristics to the specific domain we were dealing with.

Firstly, it was designed an evolutionary approach based on classic genetic algorithms (GAs). The genetic algorithm was adapted to the TAM model in order to try to find the lowest energy of the mesh, that is, the desired segmentations. The classic operators, such as the crossover and the mutation operators, were adapted to the problem, and also new *ad hoc* operators were proposed using information of the domain. Moreover, a hybrid method that uses the greedy local search and the global search of the genetic algorithm, by means of a Lamarckian strategy, was also proposed. The global search overcame the possible presence of noise in the images whereas the greedy search helped to speed up the segmentation. The hybrid combination also introduced the possibility of topological changes, provided by the greedy local search method, to perform better adjustments and segmentations in complex surfaces or even the simultaneous detection and segmentation of several objects in the scene. Both approaches were tested in several images, comparing the results with the ones obtained by the previously proposed greedy method. The results remarked the robustness of the proposed methods.

The GA approaches presented several drawbacks that we tried to overcome with other evolutionary techniques. Thus, a Differential Evolution (DE) approach was implemented that introduced some important advantages regarding the classic GA. With the new method, we gained more simplicity in the method, because the designer has to make less decisions. This new approach replaced the entire set of genetic operators by a single one included in the DE algorithm. Moreover, the GA method required different evolutionary phases with different objectives and weights of energy parameters. Meanwhile, the DE approach integrated both phases in a single one. In addition to the simplicity, the proposed method also provided a faster convergence and better results, as it was shown with representative images. As in the GA case, a hybridization of the DE approach and the greedy local search was developed, including the mentioned properties and advantages of the combination.

Another drawback of the evolutionary approaches that were proposed, which are general in single-objective optimization techniques, is the parameter tuning that has to be performed for each segmentation. We implemented a multiobjective methodology that used the SPEA2 algorithm, adapted to our application, and with the incorporation of relevant modifications in the basic algorithm. Different versions of the multiobjective method were developed: One that uses the characteristics of the adapted GA, and another improved one that included the DE algorithm in the production of new individuals instead of the GA operators.

In not very complex images all the approaches can obtain correct segmentations. However, single-objective methods need different parameter sets (energy weights) for the model in the different evolutionary phases, parameter tuning that has to be performed for each kind of image to select those suitable parameters that provide the desired result. That way, the process turns into an experimental tuning of the adequate weights for each image from multiple executions of the evolutionary processes.

That problem is overcome using the proposed multiobjective approaches. The different energy terms were synthesized in the objectives wanted by the designer. We used different objectives like In-Out (IO), Gradient Distance (GD), Distance In-Out (IOD) or IOD_i/GD as external objectives, because those adjust the contour to the objects of interest, and internal energy or β as objectives that provided smoothness and compression in the mesh. The obtained results showed correct results, even outperformed the single-objective approaches in several cases. The main drawback of the multiobjective approaches are related with the complexity and computational times.

Table 7.1 summarizes and details the main characteristics of the different evolu-

Table 7.1: Comparison among the different optimization methods.

	Greedy	**GA**	**Hybrid GA-Greedy**	**DE**	**Hybrid DE-Greedy**	**SPEA2-GA**	**SPEA2-DE**
Boundary detection	Good	Very good	Very good	Very good	Very good	Very good	Very good
Sensitivity to noise	Very high	Very low	Very low	Very low	Very low	Very low	Very low
Parameter tuning	Always	Always	Always	Always	Always	Never	Never
Detection of irregularities	Need specific parameters	Need specific parameters	Need specific parameters	Need specific parameters	Need specific parameters	Correct	Correct
Execution time	Very low	High	Medium	Medium	Medium - low	Very high	High
Topological changes	Yes	No	Yes	No	Yes	No	No

tionary proposed methods and also the greedy method previously defined.

Finally, we applied the different proposed approaches in a practical domain. The methods were used to detect and extract the optic disc in eye fundus images, that is, retinal images. We developed a large and complete set of experiments to validate the proposed methods with public databases, and which pointed out the robustness and correctness of the approaches in the selected task.

7.1 Future work

As future work, there are several tasks that can be developed to improve the methodologies.

1. One of the main drawbacks of the evolutionary approaches are the complexity and computational requirements that are needed to perform the segmentations. To speed up the different evolutionary methods we could parallelize them. In our application, the time needed for the computation of the fitness of each individual is greater with respect to the other phases of an evolutionary algorithm (basically the application of the genetic operators), because the fitness assignment implies the calculus of the different energy components. So, the parallelization can be performed easily in a multicore processor platform, with a number of threads equal to the number of cores, where each core selects

an individual from the shared population and returns the fitness to the shared population. The procedure requires a simple synchronization to execute the other phases of the evolutionary process.

2. Additionally, the link cutting procedure should be included in the multiobjective optimization approaches. In the other hybridizations the link cutting procedure was provided by the greedy local search. The main idea is to detect those nodes that can be moved to better positions and also are bad placed. This greedy local search needs a parameter set (energy weights), that implies a contradiction with the multiobjective approaches (one of the main advantages of the multiobjective approach was the no requirement of the experimentally tuned parameters). In an alternative proposal, we can experiment to apply the local search to the individuals of the Pareto Front, but using only single components of the energy. That is, we move the nondominated individuals in a greedy manner with the individual objectives, which repercutes in the progression of the whole Pareto Front.

3. We used the evolutionary approaches in a practical domain. In particular, they were used for detecting and extracting the optic disc in eye fundus images. Other practical tasks would be suitable to test the properties of the proposed methods like the localization and segmentation of the heart in short axis mid ventricular Cardiac Magnetic Resonance (CMR) images, task we are involved in and which we are performing with other segmentation techniques.

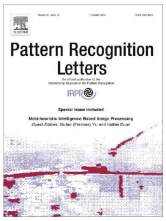
4. As a different approach to the optimization of the models, we could use Artificial Neural Networks (ANN) in the segmentation processes. The main idea would be to provide information about the image and the deformable model to the ANN with the purpose of learning the movements that should be taken for each of the nodes. Thus, once we have the ANN trained (by means of an evolutionary method), it would be considered as a "segmentation operator" which learned how to move the nodes to reach the desired segmentation. Note that this use of an ANN for the segmentation is different to a greedy minimization, as the ANN can learn to escape from the local minima as the noise presented in the images.

Appendix A

Publications

Journals

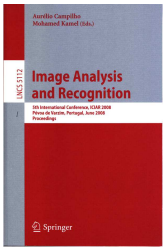
J. Novo, M. G. Penedo and J. Santos. Localisation of the optic disc by means of GA-Optimised Topological Active Nets. Image and Vision Computing, 27:1572–1584, 2009.



J. Novo, M. G. Penedo and J. Santos. Evolutionary multiobjective optimization of Topological Active Nets. Pattern Recognition Letters, 31:1781–1794, 2010.

J. Novo, N. Barreira, M. G. Penedo and J. Santos. Topological Active Volume 3D segmentation model optimized with genetic approaches. Natural Computing. 11:161–174, 2012.

Chapters in Lecture Notes Series



J. Novo, M. G. Penedo and J. Santos. Optic disc segmentation by means of GA-Optimized Topological Active Nets. Lecture Notes in Computer Science: Image Analysis and Recognition (ICIAR 2008), 5112:807–816, 2008.

J. Novo, N. Barreira, M. G. Penedo and J. Santos. Genetic approaches for the automatic division of Topological Active Volumes. Lecture Notes in Computer Science: Bioinspired Applications in Artificial and Natural Computation (IWINAC 2009), 5602:20–29, 2009.

J. Novo, J. Santos and M. G. Penedo. Optimization of Topological Active Nets with differential evolution. Lecture Notes in Computer Science: Adaptive and Natural Computing Algorithms (ICANNGA 2011), 6593:350–360, 2011.

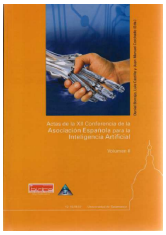
J. Novo, J. Santos and M. G. Penedo. Differential evolution optimization of 3D Topological Active Volumes. Lecture Notes in Computer Science: Advances in Computational Intelligence (IWANN 2011), 6691:282–290, 2011.

International conferences

J. Novo, J. Santos, M. G. Penedo, and A. Fernández. Optimization of topological active models with multiobjective evolutionary algorithms. In 20^{th} International Conference on Pattern Recognition (ICPR 2010), pages 2226–2229, 2010.

J. Novo, M. G. Penedo and J. Santos. Multiobjective optimization of the 3D Topological Active Volume segmentation model. In 3^{rd} International Conference on Agents and Artificial Intelligence (ICAART 2011), pages 236–241, 2011.

National conferences



J. Novo, N. Barreira, M. G. Penedo, and J. Santos. Topological active volumes optimization with genetic approaches. In Actas de la XII Conferencia de la Asociación Española para la Inteligencia Artificial, volume II, pages 41–50, 2007.

Appendix B

Resumen de la tesis. Optimización de Modelos Activos Topológicos mediante modelos evolutivos para segmentación de imagen

B.1 Introducción

En los últimos años, el procesado de imagen se ha convertido en una disciplina en pleno auge. El volumen de imágenes y vídeos a ser tratados y procesados ha ido creciendo considerablemente en la medida que la tecnología ha ido extendiendo su uso en diferentes áreas profesionales. Por ejemplo, en el campo médico, la imagen computerizada (CT) o la imagen de fondo de ojo, han aportado nueva información sobre los pacientes, información que debe ser correctamente analizada por los especialistas para el tratamiento y diagnóstico médico. Aunque es una disciplina relativamente reciente, el procesado digital de imagen ha demostrado ser muy útil en el tratamiento automático de bases de datos de imágenes. De esa manera, dicha disciplina puede ayudar a los especialistas a procesar grandes conjuntos de datos en tiempos asumibles, reemplazando los procedimientos manuales y rudimentarios usados hasta ahora, que suponían una labor larga y tediosa.

Dentro del procesamiento de imagen, la localización y segmentación de objetos y regiones es una tarea que ha ganado importancia con los años. Dicha tarea representa un paso crucial encuadrado dentro de un análisis y razonamiento de más alto nivel,

como puede ser la detección de tumores u otras patologías, dentro del ámbito médico, o la detección del área de interés en huellas dactilares o caras para análisis biométrico, entre otras disciplinas.

Existen muy diversas maneras, desarrolladas a lo largo de los años, de abordar la tarea de detección y extracción de objetos y regiones en imagen. En particular, en esta tesis doctoral nos hemos centrado en una de las disciplinas más empleadas, el uso de los *modelos deformables*. Los modelos deformables son técnicas ampliamente empleadas en análisis de imagen que han sido aplicadas con éxito en varios campos, tales como la segmentación y reconstrucción de imágenes, el reconocimiento de patrones, las simulaciones quirúrgicas o el seguimiento de estructuras en imágenes en movimiento. Los modelos deformables fueron introducidos en imágenes bidimensionales por Kass y otros [**?**] y generalizados a imágenes 3D por Terzopoulos y otros [**?**]. Posteriormente se han definido multitud de modelos deformables con características diversas. Concretamente en esta tesis, nos hemos enfocado en uno de los más completos y ventajosos para abordar la tarea de segmentación de objetos, los Modelos Activos Topológicos.

B.2 Modelos Activos Topológicos

Los Modelos Activos Topológicos (TAM), en sus 2 versiones, las Mallas Activas Topológicas [101] en 2D (Topological Active Nets, TAN) y los Volúmenes Activos Topológicos [9] en 3D (Topological Active Volumes, TAV) consisten en una variante de los modelos deformables que integra características de las técnicas de segmentación basadas en bordes y en regiones. Para ello, este modelo consta de dos tipos distintos de nodos: *nodos externos*, para el ajuste a las superficies, y *nodos internos*, para modelar la topología interna de los objetos. Los primeros usan información de bordes mientras que los segundos están relacionados con la información de regiones. Esta dualidad es la principal ventaja de este modelo frente a otros modelos ya que permite no solo el ajuste a las superficies sino también el análisis de las características internas de los objetos. Los nodos están organizados en una malla poliédrica que es deformada bajo la influencia de funciones de energía.

B.2.1 Función de energía

El modelo lleva asociado una *función de energía*, la cual es utilizada para controlar la evolución de la estructura de forma que el mínimo de la función coincide con la segmentación deseada en la imagen. Por lo tanto, dicho proceso de segmentación, usando este modelo, implica un proceso de minimización de la función de energía

asociada a la estructura. Dicha función de energía está constituida, a su vez, por diferentes componentes energéticas que implican distintas características deseadas para la segmentación. Así tenemos:

$$E(v) = \int_0^1 E_{int}(v) + E_{ext}(v) \tag{B.1}$$

donde E_{int} y E_{ext} representan la energía interna y externa del modelo asociadas a los nodos, respectivamente. Ambas componentes se descomponen internamente en nuevas subcomponentes. A grandes rasgos, ambos bloques de términos de energía controlan:

Energía interna Compuesta por términos que controlan la contracción y suavidad de la malla.

Energía externa Compuesta por términos que representan las características de la imagen que guían el proceso de ajuste.

B.2.2 Método voraz

Previamente, se ha desarrollado un método de segmentación voraz para el ajuste del modelo. Dicho método se basa en un principio sencillo: de forma iterativa analizar para cada nodo la energía que tendría si se moviera a cada una de sus posiciones vecinas, moviendo dicho nodo a la mejor posición. En caso de que ninguna posición vecina minimice la energía, dicho nodo permanece en su posición actual. De esta manera, vamos moviendo poco a poco los nodos en su vecindad reduciendo la energía, hasta que no podemos realizar movimientos en ningún nodo de la estructura, finalizando de esa manera el proceso de segmentación.

La metodología presenta, además, la opción de *cambios topológicos*, es decir, la ruptura de enlaces entre nodos para dotar a la estructura de mayor flexibilidad para un mejor ajuste en contornos complejos, como son la presencia de concavidades o huecos, por ejemplo. Una vez que la metodología no puede mover ningún nodo a una posición mejor, en ese momento es cuando analiza la posible mala ubicación de los enlaces para ver si eliminando algunos podemos realizar una mejor segmentación.

Dicho método es rápido y eficiente. Sin embargo, al ser un método de minimización local, es muy propenso a caer en mínimos locales. En los procesos de segmentación en imagen, es frecuente que las imágenes a ser analizadas presenten diferentes niveles de ruido producidos por el entorno o por las máquinas de captura de imagen. Dicho ruido es difícil que sea ignorado por un método de búsqueda local como es el método voraz, siendo altamente complicado obtener unos buenos resultados.

De esa manera, poniendo mayor énfasis en la robustez del método de segmentación, un planteamiento con métodos de búsqueda global se presenta más idóneo para obtener unos mejores resultados. La computación evolutiva ha demostrado ser un paradigma de búsqueda global que presenta buenos resultados en dominios complejos. En ese principio se basan los métodos propuestos en esta tesis, y que se detallan a continuación.

B.3 Métodos evolutivos para la optimización de los Modelos Activos Topológicos

Dadas las limitaciones comentadas del método de búsqueda local voraz para encontrar buenas segmentaciones en imágenes que presenten diversas complicaciones, en esta tesis doctoral se proponen nuevas metodologías de optimización usando computación evolutiva. Dichos métodos de búsqueda global presentan como mayor ventaja la robustez ante entornos complejos, siendo capaces de proporcionar segmentaciones aceptables.

B.3.1 Método de segmentación usando algoritmos genéticos

El primero de los métodos de segmentación propuestos hace uso de los algoritmos genéticos. Los algoritmos genéticos (AG) [37] son un método de optimización global que usa técnicas inspiradas en la evolución de las especies, usando conceptos como la herencia, mutación, cruce y selección, constituyendo un proceso de búsqueda del óptimo de la función que deseamos optimizar, en nuestro caso la energía asociada al modelo deformable.

De forma general, el proceso mediante el cual encuentra el óptimo de la función se basa en la creación sucesiva de diversas generaciones de individuos en una población dada. Dichos individuos representan cada uno de ellos una posible solución al problema, siendo codificadas en su genotipo las coordenadas de cada uno de los nodos de la estructura. Para producir una generación dada de la población se hace uso de una serie de operadores genéticos, los cuales se aplican sobre individuos de una generación para producir la siguiente. Cada individuo conlleva un *fitness*, o calidad asociada, que mide el grado de correctitud de la segmentación que representa. Para la primera generación de la población, se inicializan los individuos de forma aleatoria en el espacio de búsqueda, en nuestro caso, la imagen.

Todas las características del algoritmo genético han sido adaptadas al dominio concreto en el que hemos trabajado. En concreto, se han adaptado los operadores genéticos clásicos (cruce y mutación) así como se han propuesto nuevos operadores

específicos considerando las características de las mallas, las cuales no pueden presentar cruce de enlaces. Dichos nuevos operadores son mutación en bloque, expansión y desplazamiento, los cuales complementan a los clásicos en el proceso evolutivo.

Para el proceso de segmentación se han planteado 3 fases, de diferente relevancia, en las cuales desglosamos el proceso evolutivo de segmentación: una primera fase, o fase de localización, donde se busca centrar la población de individuos en los objetos de interés; una segunda fase, o fase de ajuste, donde se busca un ajuste más detallado en la segmentación; y una tercera fase, o fase de refinamiento, de menor relevancia donde se busca un mayor refinamiento de ajuste final y una mayor suavidad internodal en la estructura.

Además, se ha realizado una *versión híbrida* del método de búsqueda local voraz y del método de búsqueda global del algoritmo genético. De esa manera, unimos las ventajas de ambos métodos: por un lado la robustez del algoritmo genético junto con la rapidez del método voraz. Además, el propio método voraz incluye la posibilidad de cambios topológicos para realizar mejores segmentaciones en objetos de contornos complejos e incluso la segmentación de diversos objetos presentes en la escena.

La principal ventaja de los métodos propuestos reside en su robustez, comprobada en diversas imágenes que presentaban diversas complicaciones, como la presencia de ruido gaussiano, ruido local o bordes difusos. Bajo todas esas condiciones los métodos propuestos han sido capaces de proporcionar resultados aceptables. Sin embargo, los métodos propuestos presentan algunas desventajas. El más importante es la complejidad del método. Tuvimos que diseñar y emplear un conjunto de diversos operadores genéticos con la tarea de evolucionar la población a lo largo de un número considerable de generaciones hasta que conseguimos alcanzar unos resultados aceptables. Dicho proceso, aunque ha sido acelerado por el método voraz en la versión híbrida, conlleva un tiempo considerable. Además, al igual que en otros métodos de minimización de un solo objetivo, tuvimos que ajustar los parámetros que ponderan las diferentes componentes de energía hasta conseguir una configuración adecuada. Sobre esos aspectos que hemos trabajado para mejorar los métodos evolutivos propuestos.

B.3.2 Método de segmentación usando *differential evolution*

Como hemos comentado, uno de los principales problemas que presentan los métodos usando algoritmos genéticos que hemos propuesto es el uso de diversos operadores genéticos, cada uno de ellos usado según una probabilidad de aplicación que debe ser ajustada para obtener los mejores resultados posibles. Además, es conveniente obtener una mayor velocidad de convergencia de la población hacia el óptimo de-

seado, dada la lentitud de los métodos como hemos comentado. Por lo tanto, se han desarrollado una serie de mejoras en el proceso evolutivo.

Una de estas mejoras es el uso de evolución diferencial para la generación de nuevos individuos en la población del proceso evolutivo. *Differential Evolution* [77] es un método evolutivo alternativo que minimiza las decisiones del diseñador con respecto al algoritmo genético clásico. Este nuevo algoritmo sustituye todo el conjunto de operadores genéticos, previamente definidos, por uno único. Dicho operador se basa en el principio de crear nuevos individuos mediante la combinación de otros ya existentes usando una formula simple que implica cruce y mutación. Para cada individuo existente en la población, generamos un candidato según dicho nuevo operador, quedándonos en la nueva generación de la población con el mejor de ambos.

Además, al igual que lo realizado para el algoritmo genético, hemos realizado una hibridación combinando la búsqueda global de *differential evolution* junto con la búsqueda local proporcionada por el método voraz. De esa manera, además de reducir los tiempos necesarios para la segmentación, introducimos la posibilidad de eliminación de enlaces para un mejor ajuste en contornos complejos o la posibilidad de segmentar varios objetos presentes en la escena.

Además de la simplificación de operadores comentada, los resultados obtenidos han demostrado una convergencia más rápida hacia el óptimo del nuevo método evolutivo con respecto al algoritmo genético desarrollado en un primer momento. Dicho aspecto es relevante para obtener los resultados en un menor tiempo, dada la lentitud comentada de dichos métodos evolutivos de optimización.

B.3.3 Método de segmentación usando optimización multiobjetivo

Una de las principales limitaciones del algoritmo genético ha sido mejorada gracias al uso de *differential evolution*. Sin embargo, ambas metodologías, así como todos los métodos de minimización de un único objetivo, presentan una limitación con respecto a los parámetros de los componentes energéticos. En estos métodos evolutivos se asigna una calidad a cada individuo en relación con la energía asociada al modelo, energía cuyas componentes están ponderadas por diferentes parámetros. Por lo tanto, para obtener unas segmentaciones adecuadas, se deben ajustar dichos parámetros que ponderan las componentes energéticas. Dicho proceso de ajuste experimental de parámetros energéticos debe ser realizado usando cualquier método de minimización de un solo objetivo, en concreto, en nuestro caso, por el método voraz y los métodos evolutivos que usan algoritmos genéticos y differential evolution.

Los algoritmos de optimización multiobjetivo solucionan el problema del ajuste de parámetros ya que consideran la optimización de diversos objetivos en paralelo.

Dichos algoritmos normalmente trabajan en entornos con objetivos que entran en conflicto, tratando de buscar un conjunto de soluciones óptimas de compromiso llamado Frente de Pareto. Dicho conjunto está formado por soluciones no dominadas, es decir, aquellas soluciones para las cuales no hay otra que tenga todos los valores por lo menos iguales, y al menos un valor de entre los diferentes objetivos mejor. Los algoritmos evolutivos de optimización multiobjetivo [48][28] usan los principios de la computación evolutiva en la búsqueda del Frente de Pareto. En concreto, hemos desarrollado un nuevo método de segmentación que usa uno de estos algoritmos, uno de los mejores que han sido desarrollados de este tipo, el algoritmo SPEA2 [110]. En los métodos propuestos, dicho algoritmo SPEA2 fue implementado en un primer lugar usando el conjunto de operadores y características del algoritmo genético previamente desarrollado, siendo reemplazado posteriormente con las características del método basado en differential evolution.

Además de evitar todo el proceso de ajuste de los parámetros de los términos energéticos, con esta nueva metodología obtenemos un conjunto de posibles resultados finales tras el proceso de segmentación, esto es, el Frente de Pareto final respeto a los diferentes objetivos usados. De esta manera, podemos seleccionar, a posteriori, uno o varios individuos de dicho Frente, atendiendo a los criterios deseados por el usuario.

B.3.4 Aplicación práctica. Detección y extracción del disco óptico en imagen de retina

Como caso práctico, hemos empleado las técnicas evolutivas de segmentación para la tarea de localizar y segmentar el disco óptico en imágenes de retina. Los métodos han sido adaptados a este dominio específico incluyendo información del dominio.

Los métodos fueron probados en dos tipos posibles de imágenes de retina, centradas en mácula y centradas en disco óptico, al mismo tiempo que se realizó un estudio de resultados usando conjuntos públicos de imágenes de retina, aportando unos resultados satisfactorios.

B.4 Conclusiones

En esta tesis hemos diseñado e implementado diferentes metodologías para segmentación en imagen 2D y 3D. Todos los métodos usan diferentes técnicas evolutivas que mejoran los resultados obtenidos previamente, dadas las limitaciones del método voraz previamente diseñado. Estas nuevas técnicas incluyen características de la computación evolutiva así como diversas adaptaciones a los aspectos concretos

del dominio en el que estamos trabajando.

Los métodos evolutivos aportan una serie de ventajas, siendo la principal su robustez frente a diversas complicaciones que pueden aparecer en las imágenes, algo habitual cuando estamos trabajando con imágenes de un dominio real. Así, los métodos son capaces de proporcionar segmentaciones adecuadas trabajando con imágenes que presentan diversos niveles de ruido o con bordes discontinuos o difusos, como han mostrado los diversos resultados presentados en el trabajo.

En cambio, dichos métodos se caracterizan, como principal desventaja, por su lentitud. Esto es debido a la gestión de una población entera de individuos (posibles soluciones) que deben ser modificados en sucesivas generaciones hasta poder alcanzar los resultados deseados. Para ello, se han realizado diversas hibridaciones entre las técnicas evolutivas propuestas con el método voraz, el cual aporta rapidez. De esta manera conseguimos acelerar los procesos de segmentación sin perder la robustez aportada por los métodos evolutivos. Además, el método voraz incorpora el mecanismo de rupturas de enlaces, dotando de esa manera al proceso de segmentación de la posibilidad de realizar cambios topológicos. Dichos cambios aportan a la malla una gran flexibilidad para un mejor ajuste en objetos de superficies complejas (presencia de huecos o concavidades) o incluso la posibilidad de segmentar diversos objetos presentes en la escena.

Por otro lado, los métodos multiobjetivo propuestos ofrecen unas características adicionales que no aportan los métodos de optimización de un único objetivo. El método voraz y los métodos evolutivos basados en el algoritmo genético básico y el basado en *differential evolution* necesitan ajustar los parámetros energéticos para obtener una segmentación adecuada. Sin embargo, los métodos de optimización multiobjetivo omiten dicho ajuste definiendo directamente una serie de objetivos a partir de diferentes componentes de energía. Además, del Frente de Pareto final podemos extraer una o varias segmentaciones, a criterio del usuario.

Por último, se han aplicado los métodos propuestos en un dominio real, para la detección y segmentación del disco óptico en imágenes de retina, aportando unos buenos resultados.

B.5 Contribuciones

De todo el trabajo realizado se han obtenido diversas publicaciones, incluyendo publicaciones en revistas indexadas así como otras publicaciones en congresos internacionales y nacionales de reconocido prestigio.

Bibliography

[1] R. Abdel-Ghafar and T. Morris. Progress towards automated detection and characterization of the optic disc in glaucoma and diabetic retinopathy. *Informatics for Health and Social Care*, 32(1):19–25, 2007.

[2] A. S. Aguado, M. S. Nixon, and M. Montiel. Parameterizing arbitrary shapes via Fourier descriptors for evidence-gathering extraction. *Computer Vision and Image Understanding*, 69(2):202 – 221, 1998.

[3] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[4] N. C. Andreasen, R. Rajarethinam, T. Cizadlo, S. Arndt, V. W. Swayze, L. A. Flashman, D. S. O'Leary, J. C. Ehrhardt, and W. T. Yuh. Automatic atlas-based volume estimation of human brain regions from MR images. *J Comput Assist Tomogr*, 20(1):98–106, 1996.

[5] F. M. Ansia, M. G. Penedo, C. Mariño, and A. Mosquera. A new approach to active nets. *Pattern Recognition and Image Analysis*, 2:76–77, 1999.

[6] L. Ballerini. Medical image segmentation using genetic snakes. In *Proceedings of SPIE: Application and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation II*, volume 3812, pages 13–23, 1999.

[7] 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, S. Cagnoni, E. Lutton, G. Olague (Eds.)*, pages 177–194, 2007.

[8] E. Bardinet, L. D. Cohen, and N. Ayache. Tracking and motion analysis of the left ventricle with deformable superquadrics. *Medical Image Analysis*, 1:129–149, 1996.

[9] N. Barreira and M. G. Penedo. Topological Active Volumes. *EURASIP Journal on Applied Signal Processing*, 13(1):1937–1947, 2005.

[10] N. Barreira, M. G. Penedo, C. Mariño, and F. M. Ansia. Topological Active Volumes. *Lecture Notes in Computer Science: Computer Analysis of Images and Patterns*, 2756:337–344, 2003.

[11] N. Barreira, M. G. Penedo, M. Ortega, and J. Rouco. On the improvement of the Topological Active Volumes model: A tetrahedral approach. In *Third International Conference on Computer Vision Theory and Applications*, volume 1, pages 529–534, 2008.

[12] R. Bellman. *Dynamic programming.* Dover Publications, 2003.

[13] A. Blake and M. Isard. 3d position, attitude and shape input using video tracking of hands and lips. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 185–192, New York, NY, USA, 1994. ACM.

[14] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 26, Washington, DC, USA, 2003. IEEE Computer Society.

[15] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

[16] M. Bro-Nielsen. Active nets and cubes. Technical Report 13, IMM, Technical University of Denmark, 1994.

[17] E. Carmona, M. Rincón, J. Feijoó, and J. D. la Casa. Identification of the optic nerve head with genetic algorithms. *Artificial Intelligence in Medicine*, (43):243–259, 2008.

[18] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[19] T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[20] T. Chanwimaluang and G. Fan. An efficient algorithm for extraction of anatomical structures in retinal images. *IEEE International Conference on Image Processing*, 23:1093–1096, 2003.

[21] R. Chrástek, M. Wolf, K. Donath, G. Michelson, and H. Niemann. Optic disc segmentation in retinal images. In *Bildverarbeitung fur die Medizin*, pages 263–266, 2002.

[22] I. Cohen and L. D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. *Computer Vision and Image Understanding*, 63:527–541, 1994.

[23] I. Cohen and L. D. Cohen. A hyperquadric model for 2-D and 3-D data fitting. In *12th Tnternational Conference on Pattern Recognition (ICPR'94)*, pages 403–405, 1994.

[24] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[25] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[26] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[27] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.

[28] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[29] H. Delingette, S. Cotin, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Proceedings of the Computer Animation (CCA '99)*, pages 70 –81, 1999.

[30] C. Dorai and A. K. Jain. COSMOS-A representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1115–1130, October 1997.

[31] DRIVE. Digital Retinal Images for Vessel Extraction. http://www.isi.uu.nl/Research/Databases/DRIVE/.

[32] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Face recognition using active appearance models. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 581–595, London, UK, 1998. Springer-Verlag.

[33] Y. Fan, T. Jiang, and D. Evans. Volumetric segmentation of brain images using parallel genetic algorithms. *IEEE Transactions on Medical Imaging*, 21(8):904–909, 2002.

[34] V. Feoktistov. *Differential Evolution: In Search of Solutions*. Springer, NY, 2006.

[35] M. Foracchia, E. Grisan, and A. Ruggeri. Detection of optic disc in retinal images by means of a geometrical model of vessel structure. *IEEE Transactions on Medical Imaging*, 23(10):1189–1195, 2004.

[36] F. Glover and M. Laguna. *Tabu Search*. Kluwer, 1996.

[37] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[38] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 1992.

[39] S. Han, D. Goldgof, and K. Bowyer. Using hyperquadrics for shape recovery from range data. In *International Conferecence on Computer Vision (ICCV'93)*, pages 492–496, 1993.

[40] R. Hansen and G. W. Walster. *Global optimization using interval analysis*. CRC Press, 2004.

[41] A. Hill, T. F. Cootes, and C. J. Taylor. Active shape models and the shape approximation problem. In *BMVC '95: Proceedings of the 1995 British conference on Machine vision (Vol. 1)*, pages 157–166, Surrey, UK, UK, 1995. BMVA Press.

[42] J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[43] J. Holland. *Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence.* The University of Michigan Press, Ann Arbor, MI, 1992.

[44] A. Hoover and M. Goldbaum. Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. *IEEE Transactions on Medical Imaging*, 22(8):951–958, 2003.

[45] O. Ibáñez, N. Barreira, J. Santos, and M. Penedo. Genetic approaches for Topological Active Nets optimization. *Pattern Recognition*, 42(5):907 – 917, 2009.

[46] ITO. Instituto tecnológico de oftalmología. http://www.itogalicia.es/.

[47] J.-L. and Mallet. Discrete smooth interpolation in geometric modelling. *Computer-Aided Design*, 24(4):178 – 191, 1992.

[48] A. Jaimes and C. Coello. Multi-objective evolutionary algorithms: A review of the state-of-the-art and some of their applications in chemical engineering, R.G. Pandu (editor). In *Multi-Objective Optimization Techniques and Applications in Chemical Engineering, World Scientific*, pages 61–90, 2009.

[49] H. Jelinek, C. Depardieu, C. Lucas, D. Cornforth, W. Huang, and M. Cree. Towards vessel characterisation in the vicinity of the optic disc in digital retinal images. In *Proceedings of the Image and Vision Computing Conference, McCane (ed.), New Zealand 2005*, 2005.

[50] M. Kamber, R. Shinghal, D. L. Collins, G. S. Francis, and A. C. Evans. Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images. *IEEE Transactions on Medical Imaging*, 14(3):442–453, 1995.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Active contour models. *International Journal of Computer Vision*, 1(2):321–323, 1988.

[52] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 810, Washington, DC, USA, 1995. IEEE Computer Society.

[53] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[54] M. Lalonde, M. Beaulieu, and L. Gagnon. Fast and robust optic disk detection using pyramidal decomposition and Hausdorff-based template matching. *IEEE Transactions on Medical Imaging*, 20:1193–1200, 2001.

[55] F. Leitner and P. Cinquin. From splines and snakes to snake splines. In *Selected Papers from the Workshop on Geometric Reasoning for Perception and Action*, pages 264–281, London, UK, 1993. Springer-Verlag.

[56] F. Leitner, I. Marque, S. Lavallée, and P. Cinquin. Dynamic segmentation: finding the edge with snake splines. *Curves and surfaces*, pages 279–284, 1991.

[57] H. Li and O. Chutatape. Automatic location of optic disk in retinal images. In *Proceedings of the International Conference on Image Processing, Vol. 2*, pages 837–840, 2001.

[58] H. Li and O. Chutatape. Boundary detection of optic disk by a modified ASM method. *Pattern Recognition*, 36:2093–2114, 2003.

[59] D. G. Lowell. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell, MA, USA, 1985.

[60] J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, E. Fletcher, and L. Kennedy. Optic nerve head segmentation. *IEEE Transactions on Medical Imaging*, 23(2):256–264, 2004.

[61] L. MacEachern and T. Manku. Genetic algorithms for active contour optimization. In *Proc. IEEE International Symposium on Circuits and Systems*, volume 4, pages 229–232, 1998.

[62] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[63] T. Mcinerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, June 1996.

[64] F. Mendels, C. Heneghan, and J.-P. Thiran. Identification of the optic disk boundary in retinal images using active contours. *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 103–115, 1999.

[65] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: Implementation and application to stereo. *Artificial Intelligence and Computer Vision*, pages 223–236, 1991.

[66] J. M. Molina and E. J. Carmona. Localization and segmentation of the optic nerve head in eye fundus images using pyramid representation and genetic algorithms. In *Proceedings of the 4th international conference on Interplay between natural and artificial computation - Volume Part I*, IWINAC'11, pages 431–440, Berlin, Heidelberg, 2011. Springer-Verlag.

[67] J. Montagnat and H. Delingette. Volumetric medical images segmentation using shape constrained deformable models. In *First Joint Conference, Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS'97)*, pages 13–22, 1997.

[68] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: Topology, geometry and deformation. *Image and Vision Computing*, 19:1023–1040, 2001.

[69] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[70] M. Niemeijer, M. Abramoff, and B. van Ginneken. Segmentation of the optic disc, macula and vascular arch in fundus photographs. *IEEE Transactions on Medical Imaging*, 26(1):116–127, 2007.

[71] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[72] C. Ooi and P. Liatsis. Co-evolutionary-based active contour models in tracking of moving obstacles. In *International Conference on Advanced Driver Assistance Systems*, pages 58–62, 2001.

[73] A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham. Colour morphology and snakes for optic disc localisation. In *The 6th Medical Image Understanding and Analysis Conference. A Houston, R Zwiggelaar, (eds.)*, pages 21–24, 2002.

[74] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[75] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[76] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2(1):315–337, 2000.

[77] K. Price and R. Storn. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[78] K. Price, R. Storn, and J. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer - Natural Computing Series, 2005.

[79] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport. Statistical approach to segmentation of single-channel cerebral MR images. *IEEE Transactions on Medical Imaging*, 16(2):176–186, 1997.

[80] T. Robič and B. Filipič. Demo: Differential evolution for multiobjective optimization. In *In Proceedings of the 3rd International Conference on Evolutionary MultiCriterion Optimization (EMO 2005*, pages 520–533. Springer. LNCS, 2005.

[81] J. Santos, O. Ibáñez, N. Barreira, and M. Penedo. Genetic-greedy hybrid approach for topological active nets optimization. In *Adaptive and Natural Computing Algorithms - Lecture Notes in Computer Science*, volume 4431, pages 202–210, 2007.

[82] R. Séguier and N. Cladel. Genetic snakes: Application on lipreading. In *International Conference on Artificial Neural Networks and Genetic Algorithms*, 2003.

[83] R. Séguier and N. Cladel. Multiobjectives genetic snakes: Application on audio-visual speech recognition. In *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, pages 625–630, 2003.

[84] J. A. Sethian. *Level set methods: evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1996.

[85] A. Singh, D. Terzopoulos, and D. B. Goldgof. *Deformable Models in Medical Image Analysis*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1998.

[86] L. H. Staibt and J. S. Duncant. Deformable Fourier models for surface finding in 3D images. In *Visualization in Biomedical Computing (VBC'92)*, pages 90–104, 1992.

[87] G. Storvik. A bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):976–986, 1994.

[88] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, Dec. 1990.

[89] G. Székely, A. Kelemen, C. Brechbühler, and G. Gerig. Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models. *Medical Image Analysis*, 1(1):19–34, 1996.

[90] T. Tanatipanond and N. Covavisaruch. An improvement of multiscale approach to deformable contour for brain MR images by genetic algorithm. In *The 1999 International Symposium on Intelligent Signal Processing and Communication System*, pages 677–680, 1999.

[91] T. Tanatipanond and N. Covavisaruch. A multiscale approach to deformable contour for brain MR images by genetic algorithm. In *The Third Annual National Symposium on Computational Science and Engineering*, pages 306–315, 1999.

[92] G. Taubin. Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics*, 13(1):3–42, 1994.

[93] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303, 1994.

[94] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(7):703–714, 1991.

[95] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischert. Elastically deformable models. *Computer Graphics*, 21:205–214, 1987.

[96] D. Terzopoulos and R. Szeliski. *Active Vision, chapter Tracking with kalman Snakes*. MIT Press, 1992.

[97] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[98] K. Tobin, E. Chaum, V. Govindasamy, and T. Karnowski. Detection of anatomic structures in human retinal imagery. *IEEE Transactions on Medical Imaging*, 26(12):1729–1739, 2007.

[99] J. Tohka. Global optimization of deformable surface meshes based on genetic algorithms. In *Proceedings 11th International Conference on Image Analysis and Processing (ICIAP 2001)*, pages 459–464, 2001.

[100] J. Tohka and J. Mykkänen. Deformable mesh for automated surface extraction from noisy images. *International Journal of Image and Graphics*, 4(3):405–432, 2004.

[101] Y. Tsumiyama, K. Sakaue, and K. Yamamoto. Active net: Active net model for region extraction. *IPSJ SIG Notes*, 89(96):1–8, 1989.

[102] VARIA. VARPA Retinal Images for Authentication, 2008. http://www.varpa.es/varia.html.

[103] D. J. Williams and M. Shah. A Fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, Jan. 1992.

[104] O. Wirjadi. Survey of 3D image segmentation methods. Technical report, The Fraunhofer–Institut Techno- und Wirtschaftsmathematik, 2007.

[105] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[106] J. Xu and O. Chutatape. Automated detection of optic disk boundary by a new deformable model technique. *Proc. of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 6516–6519, 2005.

[107] J. Xu, E. Sung, O. Chutatape, C. Zheng, and P. Chew. Automated optic disk segmentation via a modified snake technique. In *ICARCV06*, pages 1–6, 2006.

[108] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A.Tannenbaum. A geometric snake model for segmentation of medical imagery. *Medical Imaging, IEEE Transactions on*, 16(2):199–209, 1997.

[109] A. Yuille and P. Hallinan. *Active Vision, chapter Deformable models.* MIT Press, 1992.

[110] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. In *EUROGEN 2001, Evolutionary Methods for Design, Optimisation, and Control*, pages 95–100, 2002.

[111] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.