Hybrid Modified Firefly Algorithm for Border Detection of Skin Lesions in Medical Imaging

Akemi Gálvez
Dept. of Information Science
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
galveza@unican.es

Iztok Fister
Faculty of Electrical Eng. & Comp. Sci.
University of Maribor
Maribor, Slovenia
iztok.fister2@uni-mb.si

Eneko Osaba
TECNALIA
Derio, Spain
eneko.osaba@tecnalia.com

Iztok Fister Jr.
Faculty of Electrical Eng. & Comp. Sci.
University of Maribor
Maribor, Slovenia
iztok.fister1@um.si

Javier Del Ser
TECNALIA
Derio, Spain
University of Basque Country
Bilbao, Spain
javier.delser@tecnalia.com

Andrés Iglesias
Dept. of Information Science
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
iglesias@unican.es

***Abstract*—Computerized analysis of skin lesions is an important issue in information retrieval for medical imaging, as it helps human specialists to improve their decision-making for prompt and accurate diagnosis of melanoma and other skin diseases. A relevant task in this regard is border detection, which gives valuable information about some clinical features of skin lesions. This task is typically carried out manually by the dermatologists, leading to errors inherent to subjective diagnosis. In this paper, we address this problem by applying a modification of a powerful evolutionary computation method, the firefly algorithm. The modified algorithm is hybridized with a local search procedure for better performance. Experimental results on a benchmark of medical images of skin lesions show that this method outperforms classical mathematical methods for the instances in the benchmark and is very competitive and often superior to state-of-the-art techniques in the field in terms of numerical accuracy. We conclude that the approach is very promising and can be useful in real-world medical applications where speed is not a critical factor.**

***Index Terms*—medical imaging, firefly algorithm, local search, border detection, skin lesions**

I. Introduction

A. Motivation

As the number of cases of skin cancer and other skin diseases is dramatically increasing in recent years, the diagnosis and treatment of skin lesions has become a major source of concern in current public and private medical and healthcare systems worldwide. Time is a key factor in this process: it is well-known that early detection is critical for an efficient treatment of melanoma and other malignant skin lesions, as it can reduce the mortality and morbidity of patients. Just for illustration, the five-year survival rate is about 99% for stage 0 melanoma (in situ), when the malignant tumor is still confined to the upper layers of the skin (epidermis), while it is only $7\% \sim 20\%$ for stage 4 melanoma, when the cancer has spread to other parts of the body. These figures are a clear evidence of the important role of prompt detection and accurate diagnosis.

There are several diagnostic procedures for skin lesions, the most common ones being the self-examination and the visual inspection by a specialist. However, some forms of skin cancer such as melanoma are difficult to distinguish visually from other skin lesions such as moles or dysplastic nevus. Other diagnosis procedures include the popular ABCDE method, the Menzies scale, the 7-point checklist, and different types of biopsy [9], [16]. Unfortunately, these procedures still rely on the clinical experience of the specialists, leading to diagnostic results that are somewhat subjective and can vary significantly even among experienced dermatologists.

These limitations can be alleviated by using non-invasive skin imaging techniques such as dermoscopy. These methods reduce screening errors and enhance discrimination between melanoma and other skin lesions [3]. However, they are also prone to errors due to the difficulty and subjectivity of visual perception of images, making it dependent on the proficiency of the specialist. In this sense, the use of advanced computer-aided image processing techniques is very useful as a second independent diagnosis to improve the biopsy decision-making by clinicians and for other purposes. As a consequence, computerized analysis of skin lesions is now a very active area of research in the biomedical and healthcare fields, particularly in information retrieval of medical images. The ultimate goal is to develop reliable automatic computer tools for accurate identification and classification of different skin lesions. This is also the motivation of this work. Because of limitations of space, this paper will be restricted to a particular step of the full image processing pipeline. The

978-1-7281-2153-6/19/$31.00

interested readers are kindly referred to [13], [17] for two interesting reviews on techniques and algorithms for general computer-aided diagnosis of pigmented skin lesions covering all steps in detail.

B. Image Segmentation and Border Detection

The standard approach in automatic medical image processing of skin lesions consists of three stages: 1) image segmentation; 2) feature extraction and feature selection; and 3) lesion classification. Image segmentation aims to identify the ROI (region of interest) of the skin lesion and separate it from the background. The segmentation stage is very important, not only because it is the starting point for the whole process but also because it affects the accuracy of the subsequent steps. Popular segmentation approaches include thresholding methods [5], [10], edge-based methods [2], clustering methods [19], [24], level set methods [15] and active contours [14], among others.

An important task in image segmentation is border detection, i.e., the determination of the boundary between the skin lesion and the surrounding healthy skin areas. This is a valuable input for diagnosis, as the lesion border gives information about some clinical features (for instance, irregular borders are a good indicator of possible malignant tumors). Usually, the border detection is performed manually by the dermatologists, leading to a border polyline obtained by joining consecutive feature points (acquired by visual inspection) through straight segments. This output does not represent well the real process, as the border of skin lesions rarely happens to be piecewise linear. In addition, interpolation schemes that enforces the border to pass through *all* feature points are typically affected by artifacts inherent to manual processes, such as outliers and noise in data. Under such conditions, approximation schemes (that only require the border to pass *near* the feature points) are better suited for this problem. Furthermore, approximation schemes are more advantageous in terms of accuracy, computer memory, and data storage capacity, as the border can be accurately described by a few tens of parameters even for very large collections of feature points. This is the problem addressed in this paper, explained in detail in Section II.

C. Structure of this Paper

This paper is organized as follows: Section II describes the problem to be solved, which is formulated as a continuous nonlinear optimization problem. Then, Section III describes the firefly algorithm, the evolutionary computation approach used in this work. The methodology proposed in this paper is described in detail in Section IV. Section V reports our experimental results. A comparative analysis of our method with other alternatives in the field is also discussed in that section. The paper closes with the main conclusions and some ideas for future work in the field.

II. Problem to be Solved

In this paper we assume that we are provided with a collection of feature points $\{\mathbf{\Delta}_\eta\}_{\eta=1,\ldots,\zeta}$ in $\mathbb{R}^2$ obtained by a trained dermatologist from given medical images. These feature points $\{\mathbf{\Delta}_\eta\}$ define a border between a skin lesion and the skin background,s enclosing the ROI containing the skin lesion for diagnosis. We seek to compute a model function $\mathbf{\Phi}(\xi)$ defined on a finite interval domain $[\nu_1, \nu_2]$ and approximating the data points better in the least-squares sense. Note that in this paper vectors are denoted in bold. This approximation scheme is generally more suitable for border detection by using a smooth mathematical equation than a polyline connecting the feature points through simple straight lines. We assume that $\mathbf{\Phi}(\xi)$ can be mathematically represented as a linear combination of the so-called blending functions:

$$\mathbf{\Phi}(\xi) = \sum_{\alpha=0}^{\delta} \mathbf{\Theta}_\alpha \psi_\alpha(\xi) \qquad \xi \in [\nu_1, \nu_2]. \tag{1}$$

The family of blending functions $\{\psi_\alpha(\xi)\}_\alpha$ in Eq. (1) is assumed to be linearly independent and form a basis of the vector space of functions of degree up to δ on $[\nu_1, \nu_2]$. In this paper we consider the case in which all functions $\{\psi_\alpha(\xi)\}_\alpha$ have their support on the whole domain $[\nu_1, \nu_2]$ (these functions are usually referred to as global-support functions). In practical terms, this means that the blending functions provide a global control of the shape of the approximating curve. Without loss of generality, we can assume that $[\nu_1, \nu_2]$ is the unit interval $[0, 1]$. Then, we consider the case of blending functions given by:

$$\psi_\alpha(\xi) = \binom{\delta}{\alpha} \xi^\alpha (1-\xi)^{\delta-\alpha} \tag{2}$$

which is usually called the Bernstein polynomial basis. Our goal is to obtain the global-support approximating curve that best fits these data points while keeping the number of degrees of freedom as low as possible. This leads to a difficult minimization problem involving two different (and competing) factors: the fitting error at the data points, and the number of free parameters of the model function. In this paper, the fitting error is given by the weighted least-squares functional:

$$\mathcal{E} = \sum_{\eta=1}^{\zeta} \Omega_\eta \left[\mathbf{\Delta}_\eta - \sum_{\alpha=0}^{\delta} \mathbf{\Theta}_\alpha \psi_\alpha(\rho_\eta) \right]^2 \tag{3}$$

where we need a parameter value ρ_η to be associated with each data point $\mathbf{\Delta}_\eta$. Since this functional does not consider the number of data points, we also consider the RMSE (root-mean-square error) of functional $\mathcal{E}$, given by:

$$RMSE(\mathcal{E}) = \sqrt{\frac{\mathcal{E}}{\zeta}} \tag{4}$$

The number of free parameters is computed by following a Bayesian approach (see [20] for further details). Therefore,

our optimization problem consists of minimizing the following weighted Bayesian energy functional $\mathcal{L}$:

$$\mathcal{L} = \frac{\zeta}{2}\log\left(\sum_{\eta=1}^{\zeta}\Omega_\eta\left[\mathbf{\Delta}_\eta - \sum_{\alpha=0}^{\delta}\mathbf{\Theta}_\alpha\psi_\alpha(\rho_\eta)\right]^2\right) + \frac{\zeta.\mu}{2}\left(\frac{2\delta-1}{2}\right)\log(\zeta). \quad (5)$$

Eq. (5) is comprised of two terms: the first one computes the fitting error to the data points, while the second one plays the role of a penalty term in order to reduce the degrees of freedom of the model. The penalty term also includes a real positive multiplicative factor μ used to modulate how much this term will affect the whole energy functional.

Considering the vectors $\mathbf{\Xi}_\alpha = (\psi_\alpha(\rho_1),\ldots,\psi_\alpha(\rho_\zeta))^T$, with $\alpha = 0,\ldots,\delta$, where $(.)^T$ means transposition, $\mathbf{\Xi} = (\mathbf{\Xi}_0,\ldots,\mathbf{\Xi}_\delta)$, $\mathbf{\Delta} = (\mathbf{\Delta}_1,\ldots,\mathbf{\Delta}_\zeta)$, $\mathbf{\Omega} = (\mathbf{\Omega}_1,\ldots,\mathbf{\Omega}_\zeta)$, and $\mathbf{\Theta} = (\mathbf{\Theta}_1,\ldots,\mathbf{\Theta}_\delta)^T$, Eq. (5) can be written in matricial form as:

$$\mathcal{L} = \mathbf{\Omega}.\mathbf{\Delta}^T.\mathbf{\Delta} - \mathbf{\Omega}.\mathbf{\Theta}^T.\mathbf{\Xi}^T.\mathbf{\Delta} - \mathbf{\Omega}.\mathbf{\Delta}^T.\mathbf{\Xi}.\mathbf{\Theta} + \mathbf{\Omega}.\mathbf{\Theta}^T.\mathbf{\Xi}^T.\mathbf{\Xi}.\mathbf{\Theta}. \quad (6)$$

Minimization of $\mathcal{L}$ requires to differentiate Eq. (6) with respect to $\mathbf{\Theta}$, and making it equal to zero to satisfy the first-order conditions. This leads to the following system of equations (called the *normal equations*):

$$\mathbf{\Xi}^T.\mathbf{\Xi}.\mathbf{\Theta} = \mathbf{\Xi}^T.\mathbf{\Delta} \quad (7)$$

which can be compacted as:

$$\mathcal{M}.\mathbf{\Theta} = \mathcal{R} \quad (8)$$

with $\mathcal{M} = \mathbf{\Xi}^T.\mathbf{\Xi}$ being a square matrix, and $\mathcal{R} = \mathbf{\Xi}^T.\mathbf{\Delta}$. If known values are assigned to the ρ_η, Eq. (8) is a classical linear least-squares minimization that can readily be solved by standard numerical techniques [6]. But if the ρ_η are free variables, the problem becomes more difficult. Since the blending functions $\psi_\alpha(\xi)$ are nonlinear in ρ, it is a nonlinear continuous optimization problem. It is also a multimodal problem, since there might be arguably more than one data parameterization vector leading to the optimal solution. As a conclusion, solving the parameterization problem leads to a difficult multimodal, multivariate, continuous, nonlinear optimization problem. This problem cannot be solved by using classical mathematical optimization techniques. In this paper, we will address this optimization problem by using the firefly algorithm, a powerful evolutionary computation method that has already been used for data fitting optimization in some previous works [11], [12]. In particular, we consider a hybrid approach based on some modifications of the original firefly algorithm and its subsequent combination with a local search method. These algorithms are described in detail in next section.

III. The Firefly Algorithm

The *firefly algorithm* (FFA) is a bio-inspired evolutionary computation algorithm introduced in 2008 to solve optimization problems [22], [23]. The algorithm is a computational metaphor inspired by the social flashing behavior of fireflies in nature. The basic components of the method are the variation of light intensity and the concept of attractiveness, assumed to be proportional to the brightness, associated in turn with the objective function of the given optimization problem.

A. Original Firefly Algorithm

The original firefly algorithm is based on three particular idealized rules, which account for some of the major flashing characteristics of real fireflies [22]. They are:

1) All fireflies are unisex, so that one firefly will be attracted to other fireflies regardless of their sex;
2) The degree of attractiveness of a firefly is proportional to its brightness, which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. For any two flashing fireflies, the less brighter one will move towards the brighter one. If there is not a brighter firefly, it will then move randomly;
3) The brightness or light intensity of a firefly is determined by the value of the objective function of a given problem. For instance, light intensity can simply be proportional to the value of the objective function of optimization problems.

The distance between any two fireflies i and j, at positions $\mathbf{s}_i$ and $\mathbf{s}_j$ at time instance t, respectively, can be defined as a Cartesian or Euclidean distance as follows:

$$r_{ij}^t = ||\mathbf{s}_i^t - \mathbf{s}_j^t|| = \sqrt{\sum_{k=1}^{d}(s_{i,k}^t - s_{j,k}^t)^2} \quad (9)$$

where $s_{i,k}^t$ is the k-th component of the spatial coordinate $\mathbf{s}_i^t$ of the i-th firefly and d is the number of dimensions.

In the firefly algorithm, the attractiveness function of a firefly j might be any monotonically decreasing function of the distance between the i firely and the chosen j firefly. For instance, the exponential function:

$$\beta = \beta_0 e^{-\gamma(r_{ij}^t)^2} \quad (10)$$

where r_{ij}^t is the distance defined as in Eq. (9), β_0 is the initial attractiveness at $t = 0$, and γ is an absorption coefficient at the source that controls the decrease of the light intensity.

The movement of a firefly i at time $t+1$ which is attracted by a more attractive (i.e., brighter) firefly j is governed by the following evolution equation:

$$\mathbf{s}_i^{t+1} = \mathbf{s}_i^t + \beta_0 e^{-\gamma(r_{ij}^t)^2}(\mathbf{s}_j^t - \mathbf{s}_i^t) + \upsilon\left(\sigma - \frac{1}{2}\right) \quad (11)$$

where the first term on the right-hand side is the current position of the firefly, the second term accounts for the attractiveness of the firefly to light intensity seen by other fireflies, and the third term is used for the random movement of a firefly in case there is not a brighter one. The coefficient υ is a randomization parameter, while σ is a random number uniformly distributed on the interval $[0, 1]$.

Algorithm 1 shows the pseudocode of the original firefly algorithm. The method considers an initial population $\mathcal{P}^0$ of

Require: (Initial Parameters)
Population size: $N_{\mathcal{P}}$
Maximum number of iterations: $\mathcal{T}_{max}$
Randomization parameter: υ // for eq. (11)
Initial attractiveness: β_0 // for eq. (10)
Absorption coefficient: γ // for eq. (10)
Dimension of the problem: d
Fitness function: $\varphi(\mathbf{s})$

1: $t \leftarrow 0$
2: Initialize the firefly population $\mathcal{P}^0 = \{\mathbf{s}_i^0\}_{i=1,\ldots,N_{\mathcal{P}}}$
3: **while** $t < \mathcal{T}_{max}$ **do**
4: Generate $\upsilon^t \leftarrow$ *UpsilonNew*()
5: Evaluate $\mathbf{s}_i^t \leftarrow$ *EvaluateFFA*$(\mathcal{P}^t, \varphi)$
6: Sort the $\mathbf{s}_i^t$ according to φ
7: $\mathbf{s}^* \leftarrow$ *Best*$(\mathcal{P}^t)$
8: Compute $\mathbf{s}^{t+1}$ // eqs. (9)-(11)
9: $t \leftarrow t + 1$
10: **end while**
11: **return** $\mathbf{s}^*$

Algorithm 1: Firefly algorithm pseudocode

$N_{\mathcal{P}}$ individuals $\{\mathbf{s}_i^0\}_{i=1,\ldots,N_{\mathcal{P}}}$ initialized randomly. This population size is maintained along the iterations. The firefly search is performed within the while loop (lines 3-10 in Algorithm 1), with the following steps: firstly, the function *UpsilonNew*() modifies the value of parameter υ (line 4), although this step is optional in the algorithm, and a constant value can be considered instead. Then, the function *EvaluateFFA* evaluates the quality of the solution (line 5). All solutions at time t are sorted according to the fitness function (line 6) and the best value is selected and assigned to the global best $\mathbf{s}^*$ (line 7). The new positions are computed according to the Eqs. (9)-(11), where the fireflies move towards the more attractive individuals of the swarm (line 8). The process is repeated iteratively until the maximum number of iterations is reached. Finally, the best individual at final iteration is selected as the solution of the optimization problem (line 11).

B. Hybrid Modified Firefly Algorithm

Since its publication, the firefly algorithm has been successfully applied to several engineering and industrial problems. Also, several modifications and improvements on the original version have been developed. Another promising area of research is given by its hybridization with local search procedures. In this paper, we follow both approaches simultaneously and enrich the firefly algorithm with three additional features: the use of adaptation strategies on the control parameters, a new population model based on elitism, and its hybridization with a local search heuristics.

The first modification consists of introducing adaptation strategies on two control parameters: the randomization parameter υ, and the light absorption coefficient γ. For parameter υ it is advisable to start with relatively large values with the aim to promote extensive exploration at earlier stages of the execution, and gradually decrease it to values near to 0, so as the system performs intensive exploitation in the neighborhood of the optima. In this paper, we consider an adaptive perturbation driven by: $\upsilon^{t+1} = \upsilon^t/\sqrt{t}$ for a problem-dependent initial parameter value υ^0 to be chosen. The parameter γ is also modified according to the law: $\gamma^{t+1} = \gamma^t/t^2$, for an initial value γ^0 selected to be small, so that the algorithm quickly reaches the asymptotic behavior for $\gamma \to 0$, meaning $\beta \to \beta_0$ according to Eq. (10); in other words, the attractiveness remains mostly constant throughout the search space.

The second modification is applied on the population. In the original firefly algorithm, the initial population $\mathcal{P}^0$ of $N_{\mathcal{P}}$ individuals is entirely replaced in each generation, thus lacking some valuable features in evolutionary algorithms such as the selection pressure for survival of individuals. In fact, even the best firefly in each generation is not necessarily preserved for the next generation. In our approach, the initial population of $N_{\mathcal{P}}$ individuals is divided into three groups of sizes $N_{\mathcal{P}_j}$, $(j = 1, 2, 3)$, so that $N_{\mathcal{P}_1} + N_{\mathcal{P}_2} + N_{\mathcal{P}_3} = N_{\mathcal{P}}$. The first group is comprised by the best parents (according to the fitness), so that the best $N_{\mathcal{P}_1}$ individuals are then preserved for the next generation as members of the first group. This group ensures elitism and selection pressure, since the best individuals are always preserved over the time and the selection is based on the survival of the fittest. The second group is obtained by applying a mutation operator onto the members of the first group, so as to enforce exploitation of current good solutions. In this work, we consider a single-point, inductive uniform mutation operator that introduces random perturbations in the form of an additive Gaussian white noise of mean zero and variance θ on a randomly chosen coordinate k of the firefly $\mathbf{s}_i$, while all other coordinates remain unaltered. Finally, the third group consists of new random individuals injected at every iteration to promote diversity and hence, intensive exploration of the whole search space. This feature is particularly desirable at initial stages of the execution, while its relevance decreases over the iterations. For this reason, the population size of this group, $N_{\mathcal{P}_3}$, is dynamically modified over the iterations as: $N_{\mathcal{P}_3}^{t+1} = N_{\mathcal{P}_3}^t \left(1 - \frac{t-1}{\mathcal{T}_{max}}\right)^2$ for a given $N_{\mathcal{P}_3}^0$. At each iteration t, the three lists are combined altogether and all individuals are ranked according to the fitness. Then, the three new groups are generated and the whole process is repeated.

Finally, this modified scheme is improved by its hybridization with a local search procedure. In this work, we use a direct local search method called *Mesh Adaptive Direct Search* (MADS), originally proposed in [4] for constrained nonlinear optimization. This method extends the generalized pattern search (GPS) family of search methods by allowing local exploration, called *polling*, in an asymptotically dense set of directions in the search space. Instead of restricting the local exploration of the space of variables to a finite number of directions, it promotes a broader exploration while also showing a very good scalability. As a consequence, MADS performs pretty well in problems where both GPS and Melder-

Require: (Initial Parameters)
Initial search point: x_0
Fitness functions: φ
Maximum number of iterations: $\mathcal{K}_{max}$
Initial mesh size parameter: Δ_0^m
Initial poll size parameter: Δ_0^p
Positive integer: w^+
Negative integer: w^-
Fixed rational number: τ

1: $k \leftarrow 0$
2: **while** $k < \mathcal{K}_{max}$ **do**
3: Perform *Search* and possibly *Poll* until an improved mesh point x_{k+1} is found on the mesh M_k
4: Optional *Search*: evaluate φ on a finite subset of trial points on the mesh M_k
5: Local *Poll*: evaluate φ on the frame p_k
6: Update Δ_{k+1}^m as: $\Delta_{k+1}^m = \tau^{w_k}\Delta_k^m$ where $w_k \in \{0, 1, \ldots, w^+\}$ if an improved mesh point is found, and $w_k \in \{w^-, w^- - 1, \ldots, -1\}$ otherwise
7: Update Δ_{k+1}^p such that: $\Delta_k^m \leq \Delta_k^p$, $\forall k$
8: $k \leftarrow k + 1$
9: **end while**
10: **return** $x_{\mathcal{K}_{max}}$

Algorithm 2: Mesh adaptive direct search pseudocode

Mead methods are known to stagnate.

MADS operates in iterative fashion: at each iteration a finite number of trial points are generated and the infeasible trial points are discarded. The fitness function values at the feasible trial points are compared with the current best feasible objective value obtained so far. Each of these trial points lies on the current mesh, M_k, formed by the finite set of n_D directions $D \subset \mathbb{R}^n$ scaled by a *mesh size parameter* $\Delta_k^m \in \mathbb{R}_+$. The mesh is formed by adding the current point to a scalar multiple of a set of vectors called a pattern. If the pattern search algorithm finds a point in the mesh improving the objective function at the current point, the new point becomes the current one at next step of the algorithm. Such a trial point is called an *improved mesh point*, and the iteration is called a *successful iteration*. Each iteration is divided into two steps: the *search step* and the *poll step*. The former allows evaluation of the objective function at any finite number of mesh points. The search is called empty when no trial points are considered. When an improved mesh point is generated, the iteration may stop, or it may continue if there is a reasonable chance to find a better improved mesh point. In any case, the iteration starts with the best feasible solution achieved so far and with a mesh size parameter equal to or larger than Δ_k^m, in order to speed up convergence. Whenever the search step fails and no improved mesh point is attained, the second step, called poll step, is invoked. In this step, the mesh size parameter is reduced to increase the mesh resolution so that trial points closer to the current best solution are evaluated. To this aim, MADS includes a new parameter, Δ_k^p, called *poll size parameter*, which defines the distance from the trial points generated by the poll step to the current best solution. The set of trial points considered during the poll step is called a *frame*. The MADS frame is constructed using the current best solution x_k (called the *frame center*) and the poll and mesh size parameters to obtain a positive spanning set of directions D_k. Then, at iteration k the MADS frame is given by: $p_k = \{x_k + \Delta_k^m d; d \in D_k\}$ where d is a nonnegative number combination of the directions D_k, and $\Delta_k^m ||d|| \leq \Delta_k^p max\{||d'||; d' \in D\}$, where $D \subset \mathbb{R}^n$. The corresponding pseudocode is shown in Algorithm 2.

IV. Proposed Methodology

A. Our Method

Our method consists of applying the hybrid modified firefly algorithm described in previous section to the border detection problem described in Sect. II. Solving this problem requires to define some important issues. Firstly, we need an adequate representation of the problem. Each firelfy, representing a potential solution, corresponds to a parametric vector of the form: $\mathbf{s}^i = (\rho_1^i, \rho_2^i, \ldots, \rho_\zeta^i) \in [0,1]^\zeta$, $(i = 1, \ldots, N_\mathcal{P})$ where the $\{\rho_k^i\}_{k=1,\ldots,\zeta}$ are strictly increasing parameters. These parametric vectors are initialized with random values, which are then sorted in increasing order. Secondly, we consider the fitness functions described by the error functionals in (3)-(5), which represent a trade-off between the conflicting goals of the least-squares minimization of the fitting error at data points and the minimization of the number of free variables of the problem. Application of our modified firefly method according to Algorithm 1 for a given number of iterations $\mathcal{T}_{max}$ yields new fireflies representing the potential solutions of this optimization problem. The method is enhanced by its hybridization with the MADS local search procedure for a maximum number of iterations $\mathcal{K}_{max}$, according to Algorithm 2. Of course, both methods are applied for a given set of parameter values (see our discussion on the parameter tuning below). The firefly with the best global value for our fitness function is taken as the final solution of our minimization problem. The output of our method is the free-form parametric curve given by (1)-(2) that approximates the data points better according to the least-squares error while minimizing the number of free parameters of the problem.

B. Parameter Tuning and Implementation Details

It is well-known that metaheuristic techniques require proper parameter tuning for good performance [7]. Unfortunately, the choice of suitable parameter values is also strongly problem-dependent. Consequently, our choice is mostly based on empirical results. For the population size we set the value $N_\mathcal{P} = 100$, as larger populations take longer without changing our results significantly. This initial population is distributed as: $N_{\mathcal{P}_1} = N_{\mathcal{P}_2} = 35$ and $N_{\mathcal{P}_3}^0 = 3$. The method is executed for $\mathcal{T}_{max}$ iterations. In our simulations, we found that $\mathcal{T}_{max} = 20,000$ is enough to reach convergence in all cases. The parameters β_0 and γ are set to 1 and 0.5,

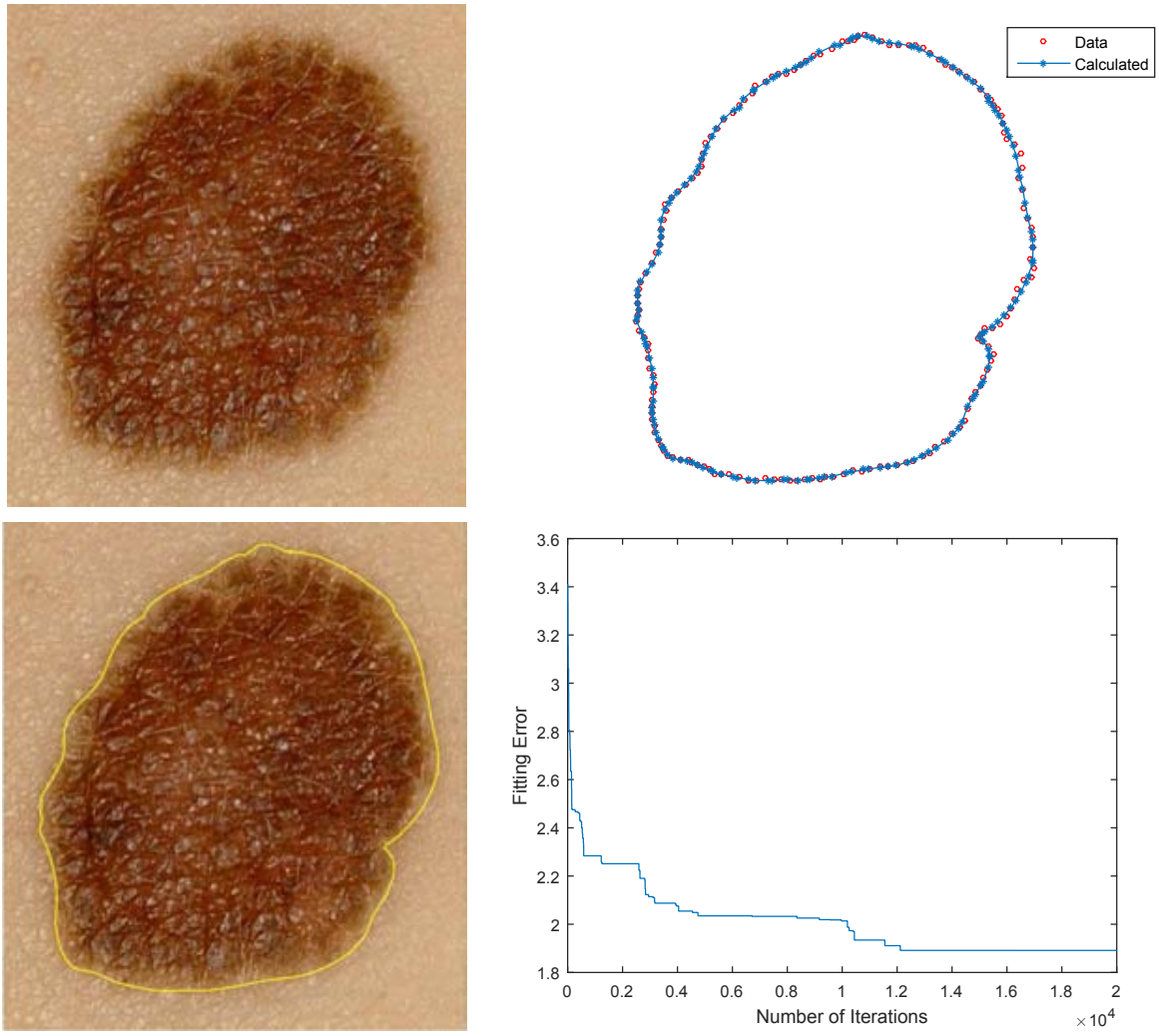

Fig. 1. Example III: (top-left) Original lesion image; (top-right) best reconstructed curve and data points; (left-bottom) reconstructed curve and lesion image; (right-bottom) convergence diagram.

TABLE I
COMPARATIVE RESULTS OF OUR METHOD WITH STATE-OF-THE-ART APPROACHES IN THE LITERATURE.

Example	*# Data points*	*Uniform*	*Chord-length*	*Centripetal*	*Thresholding*	*Clustering*	*Our method*
I	223	3.4726E−1	3.4726E−1	3.4726E−1	3.4726E−1	3.4726E−1	**3.4726E−1**
II	77	2.3525E−1	2.0219E−1	1.9832E−1	1.6318E−1	1.5494E−1	**1.3243E−1**
III	162	3.6322E−1	2.9177E−1	2.8351E−1	2.1166E−1	2.1397E−1	**1.8913E−1**
IV	220	3.4592E−1	2.5924E−1	2.6102E−1	2.1358E−1	2.0709E−1	**1.8215E−1**
V	149	4.3651E−1	4.0547E−1	3.9833E−1	3.6815E−1	3.7626E−1	**3.2537E−1**
VI	213	3.7615E−1	3.4423E−1	3.4681E−1	3.1244E−1	**2.9275E−1**	2.9323E−1
VII	96	1.6013E−1	1.2814E−1	1.3207E−1	9.2512E−2	9.2753E−2	**8.7354E−2**
VIII	113	3.0392E−1	2.6322E−1	2.6019E−1	2.3907E−1	2.4449E−1	**2.1163E−1**
IX	162	2.9667E−1	2.5291E−1	2.5488E−1	**2.2904E−1**	2.5306E−1	2.3101E−1
X	155	2.4893E−1	2.1133E−1	2.0426E−1	1.8609E−1	1.8214E−1	**1.7558E−1**

respectively. Finally, the parameter υ^0 take values within the interval $[0.5, 2]$, depending on the example. Unfortunately, a detailed analysis about how all parameters affect the method is not possible here because of limitations of space.

Regarding the implementation, our computational work has been performed on a personal PC with a 2.8 GHz Intel Core i7 processor and 16 GB of RAM. The source code has been implemented by the authors in the programming language of

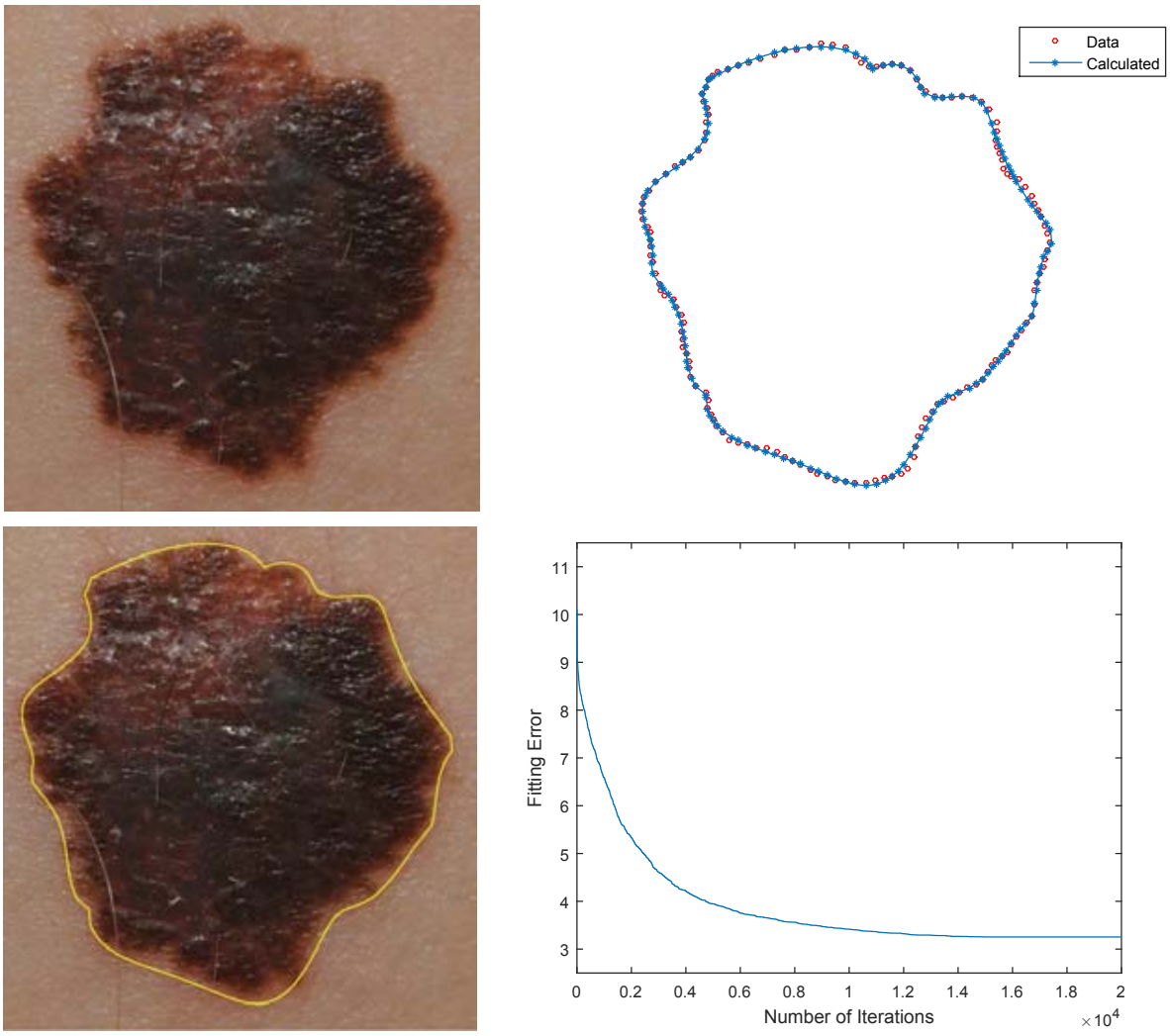

Fig. 2. Example V: (top-left) Original lesion image; (top-right) best reconstructed curve and data points; (left-bottom) reconstructed curve and lesion image; (right-bottom) convergence diagram.

the popular numerical program *Matlab*, version 2015b.

V. Experimental Results

A. Benchmark and Results

Our method has been applied to several examples of skin lesion images, obtained from the digital image archive of the Department of Dermatology of the University Medical Center of Groningen (The Netherlands). In this paper we restrict to a benchmark of ten medical images, labelled from Example I to Example X and reported in Table I (see details below). Only two examples of our benchmark (Examples III and V) are graphically displayed in this paper owing of space limitations. They are shown in Figs. 1 and 2, respectively. Both figures show the original skin lesion image (top-left), the best reconstructed border curve along with the original and the reconstructed data points (top-right), the combination of the two previous images for better visualization (bottom-left) and the convergence diagram (bottom-right). From the pictures we can see that the method obtains a very good fitting of the data points for both examples. This fact is clearly visible in the top-right pictures, where the original data points (as red empty circles) and the reconstructed curve (in blue solid line) with filled diamond symbols for the corresponding reconstructed points are displayed. Note the very good matching between the original and the reconstructed points for both examples. Similarly, good visual results have also been obtained for the other eight examples in the benchmark.

Our visual observations of this good fitting are confirmed by our numerical results, reported in Table I. For each example in the benchmark (arranged in rows), the table shows the fitting error values obtained with our method (last column) for the number of original data points indicated in the second column.

The main limitation of our method is the computing time. Although the CPU time depends on the particular example

under analysis and many other factors, our method is not really fast, as it requires typically more than 10,000 iterations to converge, i.e., several minutes for an individual execution. These CPU times compare well with other approaches in the field, but they become unacceptable for real-time medical applications or those where speed is a critical factor.

B. Comparative Analysis

We have also carried out a comparative analysis of our method with other approaches described in the literature. In particular, we consider three classical data parameterization schemes (uniform, chord-length and centripetal; see [8], [18] for details) and two state-of-the-art approaches: thresholding [21] and clustering [5]. The corresponding fitting error results are shown in columns 3–7 of Table I. For each example, the best fitting results are highlighted in bold for easier identification. Our comparative results show that our method outperforms all alternative methods in our comparison for most (but not all) of the examples in the benchmark. The only exceptions are Examples VI and IX, where the clustering and the thresholding methods prevail, respectively. We remark however that the fitting errors of our method for those examples, although slightly worse than the best values, are still very close to them. This means that our method is very competitive with respect to state-of-the-art techniques in the field for all instances in our benchmark.

VI. Conclusions and Future Work

In this paper, we applied a new hybrid modified firefly algorithm to compute the border curve of skin lesions from data points extracted from medical images. Experimental results on a benchmark of medical images show that our method outperforms classical mathematical methods and is very competitive and often superior to state-of-the-art techniques in the field in terms of numerical accuracy. The limiting factor is that the method is not really fast. We conclude that the approach is very promising and can be useful in real-world medical applications where speed is not a critical factor.

As a next step, we plan to analyze recent methods based on fusion thresholding [1] to obtain the feature points without human intervention so that our method is integrated in a fully automatic pipeline. Reducing the computing times of our method is also part of our plans for future work in the field.

Acknowledgment

Akemi Gálvez and Andrés Iglesias acknowledge the financial support from the project PDE-GIR of the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 778035, and from the the Spanish Ministry of Science, Innovation and Universities (Computer Science National Program) under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE). Iztok Fister and Iztok Fister Jr. acknowledge the financial support from the Slovenian Research Agency (Research Core Founding No. P2-0041 and P2-0057). Eneko Osaba and Javier Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program.

References

[1] Q. Abbas, M. E. Celebi, I. F. Garcia, M. Rashid: Lesion Border Detection in Dermoscopy Images Using Dynamic Programming, Skin Research and Technology, 17(1), 91–100 (2011).

[2] Abbas, A.A., Guo, X., Tan, W.H., Jalab, H.A.: Combined spline and B-spline for an improved automatic skin lesion segmentation in dermoscopic images using optimal color channel. *Journal of Medical Systems*, **38**, 80–80 (2014).

[3] Argenziano, G., Soyer, H.P., De Giorgi, V.: *Dermoscopy: A Tutorial*. EDRA Medical Publishing & New Media (2002).

[4] Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, **17**(1) (2006) 188-217.

[5] M. E. Celebi, H. Iyatomi, H., Schaefer, G., Stoecker, W.V.: Lesion border detection in dermoscopy images. *Computerized Medical Imaging and Graphics*, **33**(2) 148–153 (2009).

[6] Dierckx, P.: *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford (1993)

[7] Engelbrecht, A.P.: *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons, Chichester, England (2005).

[8] Farin, G.: *Curves and surfaces for CAGD (5th ed.)*. Morgan Kaufmann, San Francisco (2002).

[9] Friedman, R.J., Rigel, D.S., Kopf, A.W.: Early detection of malignant melanoma: The role of physician examination and self-examination of the skin. *Cancer Journal for Clinicians*, **35**(3), 130–151 (1985).

[10] Garnavi, R. Aldeen, M., Celebi, M.E., Varigos, G., Finch, S.: Border detection in dermoscopy images using hybrid thresholding on optimized color channels. *Computerized Medical Imaging and Graphics*, **35**(2), 105–115 (2011).

[11] Iglesias, A., Gálvez, A.: Memetic firefly algorithm for data fitting with rational curves. In: Proc. of IEEE Congress on Evolutionary Computation, CEC'2015, IEEE, 507–514 (2015).

[12] Iglesias, A., Gálvez, A.: New memetic self-adaptive firefly algorithm for continuous optimisation. *Int. J. of Bio-inspired Computation*, **8**(5), 300–317 (2016).

[13] Korotkov, K., Garcia, R.: Computerized analysis of pigmented skin lesions: a review. *Artificial Intelligence in Medicine*, **56**, 69–90 (2012).

[14] Ma, Z., Tavares, J.M.: A novel approach to segment skin lesions in dermoscopic images based on a deformable model. *IEEE Journal of Biomedical and Health Informatics*, **20**, 615–623 (2016).

[15] Machado, D.A., Giraldi, G., Novotny, A.A.: Multi-object segmentation approach based on topological derivative and level set method. *Integrated Computer-Aided Engineering*, **18**, 301–311 (2011).

[16] Nachbar, F., Stolz, W., Merkle, T., Cognetta, A.B., Vogt, T., Landthaler, M., Bilek, P., Braun-Falco, O., Plewig, G.: The ABCD rule of dermatoscopy. High prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal American Academy of Dermatology*, **30**(4), 551–559 (1994).

[17] Pathan, S., Prabhu, K.G., Siddalingaswamy, P.C.: Techniques and algorithms for computer aided diagnosis of pigmented skin lesions – a review. *Biomedical Signal Processing and Control*, **39**, 237–262 (2018).

[18] Piegl, L., Tiller, W.: *The NURBS Book*, Springer Verlag, Berlin Heidelberg (1997).

[19] Schmid, P.: Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Transactions on Medical Imaging*, **18**(2), 164–171 (1999).

[20] Schwarz, G.E.: Estimating the dimension of a model. *Annals of Statistics*, **6**(2) (1978) 461-464.

[21] Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, **13**, 146–165 (2004).

[22] Yang, X.S.: Firefly algorithms for multimodal optimization. Lectures Notes in Computer Science, 5792,169–178 (2009).

[23] Yang, X.S.: Firefly algorithm, stochastic test functions and design optimisation. Int. Journal of Bio-Inspired Computation, 2(2) 78–84 (2010).

[24] Zhou, H., Schaefer, G., Sadka, A., Celebi, M.E.: Anisotropic mean shift based fuzzy c-means segmentation of dermoscopy images. *IEEE Journal of Selected Topics in Signal Processing*, **3**(1) 26–34 (2009).