Automatic Fitting of Feature Points for Border Detection of Skin Lesions in Medical Images with Bat Algorithm

Akemi Gálvez[1,2], Iztok Fister[3], Iztok Fister Jr.[3], Eneko Osaba[4], Javier Del Ser[4,5,6], Andrés Iglesias[1,2,†]

[1]Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan
[2]Universidad de Cantabria, Avda. de los Castros, s/n, 39005, Santander, Spain
[3]University of Maribor, Smetanova, Maribor, Slovenia,
[4]TECNALIA, Derio, Spain
[5]University of the Basque Country (UPV/EHU), Bilbao, Spain
[6]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain
[†]Corresponding author. Email: iglesias@unican.es
Website: http://personales.unican.es/iglesias

Abstract. This paper addresses the problem of automatic fitting of feature points for border detection of skin lesions. This problem is an important task in segmentation of dermoscopy images for semi-automatic early diagnosis of melanoma and other skin lesions. Given a set of feature points selected by a dermatologist, we apply a powerful nature-inspired metaheuristic optimization method called bat algorithm to obtain the free-form parametric Bézier curve that fits the points better in the least-squares sense. Our experimental results on two examples of skin lesions show that the method performs quite well and might be applied to automatic fitting of feature points for border detection in medical images.

Keywords: Computational intelligence, medical images, skin lesion, border detection, nature-inspired metaheuristic techniques, bat algorithm

1 Introduction

Early detection and efficient treatment of skin lesions have become major concerns in current health systems worldwide. While many skin lesions are simply disturbing because of aesthetical reasons (e.g., moles, nevus), others can be very dangerous (e.g., melanomas). According to the *World Cancer Report 2014*, melanoma is the most frequent and most dangerous type of skin cancer, with an increasing number of cases and casualties every year.

Visual inspection by a specialist is the most common diagnostic procedure. However, it is difficult to distinguish the melanoma 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. However, these procedures rely heavily on human intervention, leading to diagnostic results that can vary even among experienced dermatologists.

Image-based methods are gaining traction in the field in recent years. The most classical one is dermoscopy, an epiluminescence microscopy diagnostic technique that uses a device called dermatoscope to distinguish between benign and malignant (cancerous) skin lesions, especially melanoma. Typically, the device uses a liquid medium or polarized light to cancel out skin surface reflections. During examination, the images are digitally captured in an process called digital epiluminescence dermoscopy. Dermoscopy is more precise than naked eye examination in about 20% for sensitivity (detection of melanomas) and about 10% for specificity (percentage of non-melanomas correctly diagnosed as benign). It also reduces screening errors as it enhances discrimination between real melanoma and other skin lesions [2]. However, dermoscopy is also prone to errors due to the difficulty and subjectivity of visual interpretation of images, and it is strongly dependent on the proficiency of the specialist.

Some computer-aided techniques have been developed for automatic analysis of dermoscopy images. An important step in the process is image segmentation, where the lesion is roughly separated from the background skin for better identification and classification of the skin lesion. Popular approaches for this problem include thresholding methods [4, 13], edge-based methods [1], clustering methods [25, 35], level set methods [22] and active contours [21], among others.

An important task in segmentation is the border detection of the skin lesion from the image. This is a valuable source of information for accurate diagnosis, as several clinical features can be computed directly from the detected border. Until recently, the border detection was handled manually by dermatologists by clicking with the mouse on different parts of the image on a computer screen to obtain an initial collection of feature points joined with linear segments. However, the resulting polyline is not well suited for this process, as the border of skin lesions rarely happens to be piecewise linear, but smooth. Given the input data, parametric approximation schemes are clearly better suited for this task.

This paper addresses the problem of automatic fitting of feature points for border detection of skin lesions from dermoscopy images. Given a set of feature points selected by a specialist, the method applies a powerful nature-inspired metaheuristic optimization method called bat algorithm to obtain an accurate approximation of such feature points by using free-form parametric curves (in particular, polynomial Bézier curves). This problem can be mathematically expressed as a least-squares minimization problem. However, traditional mathematical techniques fail to solve the general case, as it leads to a difficult multimodal, multivariate, continuous nonlinear optimization problem. Our method solves this minimization problem without any previous knowledge about the problem beyond the data points. Two illustrative examples of skin lesions are used to discuss the performance of the method. Our results show that the method performs quite well and can be applied for semi-automatic segmentation of medical images. The structure of this paper is as follows: Sect. 2 describes some previous work on parametric curve approximation. Then, Sect. 3 describes the bat algorithm. Our proposed method is described in detail in Sect. 4, and the experimental results are briefly discussed in Sect. 5.

2 Previous Work

Mathematically, the problem of data approximation with parametric curves can be formulated as an optimization problem, mostly solved through numerical procedures [5, 18]. Other methods use error bounds [23], curvature-based squared distance minimization [27], or dominant points [24]. These methods require some particular constraints (such as high differentiability, noiseless data) which are not commonly met in real-world applications. Later on, it was shown that artificial intelligence techniques (mostly based on neural networks) can improve such results, either alone [14, 15, 20], combined with partial differential equations [3], generalized to functional networks [16], combining functional networks and genetic algorithms [12], or using support vector machines [17].

Other approaches are based on computational intelligence, mainly nature-inspired metaheuristic techniques [6, 19, 28]. Metaheuristic approaches applied to this problem include genetic algorithms [33, 34], particle swarm optimization [7], artificial immune systems [11, 26], firefly algorithm [8, 9] and memetic approaches [10]. However, these methods are designed for explicit curves and are not applicable to the parametric case. In this paper we aim at filling this gap by applying a method called bat algorithm, described in next section.

3 The Bat Algorithm

The *bat algorithm* is a bio-inspired swarm intelligence algorithm originally proposed by Xin-She Yang in 2010 to solve optimization problems [29–32]. The algorithm is based on the echolocation behavior of microbats, which use a type of sonar called *echolocation*, with varying pulse rates of emission and loudness, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. The idealization of the echolocation of microbats is as follows:

1. Bats use echolocation to sense distance and distinguish between food, prey and background barriers.
2. Each virtual bat flies randomly with a velocity $\mathbf{v}_i$ at position (solution) $\mathbf{x}_i$ with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. As it searches and finds its prey, it changes wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r, depending on the proximity of the target.
3. It is assumed that the loudness will vary from an (initially large and positive) value A_0 to a minimum constant value A_{min}.

Some additional assumptions are advisable for further efficiency. For instance, we assume that the frequency f evolves on a bounded interval $[f_{min}, f_{max}]$. This means that the wavelength λ is also bounded, because f and λ are related to each other by the fact that the product $\lambda.f$ is constant. For practical reasons, it is also convenient that the largest wavelength is chosen such that it is comparable to the size of the domain of interest (the search space for optimization problems). For simplicity, we can assume that $f_{min} = 0$, so $f \in [0, f_{max}]$. The rate of pulse

Require: (Initial Parameters)
Population size: $\mathcal{P}$; Maximum number of generations: $\mathcal{G}_{max}$; Loudness: $\mathcal{A}$
Pulse rate: r ; Maximum frequency: f_{max} ; Dimension of the problem: d
Objective function: $\phi(\mathbf{x})$, with $\mathbf{x} = (x_1, \ldots, x_d)^T$; Random number: $\theta \in U(0,1)$

$g \leftarrow 0$
Initialize the bat population $\mathbf{x}_i$ and $\mathbf{v}_i$, $(i = 1, \ldots, n)$
Define pulse frequency f_i at $\mathbf{x}_i$
Initialize pulse rates r_i and loudness $\mathcal{A}_i$
while $g < \mathcal{G}_{max}$ **do**
  for $i = 1$ **to** $\mathcal{P}$ **do**
    Generate new solutions by using eqns. (1)-(3)
    if $\theta > r_i$ **then**
      $\mathbf{s}^{best} \leftarrow \mathbf{s}^g$ //select the best current solution
      $\mathbf{ls}^{best} \leftarrow \mathbf{ls}^g$ //generate a local solution around $\mathbf{s}^{best}$
    end if
    Generate a new solution by local random walk
    if $\theta < \mathcal{A}_i$ *and* $\phi(\mathbf{x_i}) < \phi(\mathbf{x}^*)$ **then**
      Accept new solutions, increase r_i and decrease $\mathcal{A}_i$
    end if
  end for
  $g \leftarrow g + 1$
end while
Rank the bats and find current best $\mathbf{x}^*$
return $\mathbf{x}^*$

Algorithm 1: Bat algorithm pseudocode

can simply be in the range $r \in [0, 1]$, where 0 means no pulses at all, and 1 means the maximum rate of pulse emission. With these idealized rules indicated above, the basic pseudo-code of the bat algorithm is shown in Algorithm 1. Basically, the algorithm considers an initial population of $\mathcal{P}$ individuals (bats). Each bat, representing a potential solution of the optimization problem, has a location $\mathbf{x}_i$ and velocity $\mathbf{v}_i$. The algorithm initializes these variables with random values within the search space. Then, the pulse frequency, pulse rate, and loudness are computed for each individual bat. Then, the swarm evolves in a discrete way over generations, like time instances until the maximum number of generations, $\mathcal{G}_{max}$, is reached. For each generation g and each bat, new frequency, location and velocity are computed according to the following evolution equations:

$$f_i^g = f_{min}^g + \beta(f_{max}^g - f_{min}^g) \tag{1}$$

$$\mathbf{v}_i^g = \mathbf{v}_i^{g-1} + \left[\mathbf{x}_i^{g-1} - \mathbf{x}^*\right] f_i^g \tag{2}$$

$$\mathbf{x}_i^g = \mathbf{x}_i^{g-1} + \mathbf{v}_i^g \tag{3}$$

where $\beta \in [0, 1]$ follows the random uniform distribution, and $\mathbf{x}^*$ represents the current global best location (solution), which is obtained through evaluation of the objective function at all bats and ranking of their fitness values. The

superscript $(.)^g$ is used to denote the current generation g. The best current solution and a local solution around it are probabilistically selected according to some given criteria. Then, search is intensified by a local random walk. For this local search, once a solution is selected among the current best solutions, it is perturbed locally through a random walk of the form: $\mathbf{x}_{new} = \mathbf{x}_{old} + \epsilon \mathcal{A}^g$, where ϵ is a uniform random number on $[-1, 1]$ and $\mathcal{A}^g =< \mathcal{A}_i^g >$, is the average loudness of all the bats at generation g. If the new solution achieved is better than the previous best one, it is probabilistically accepted depending on the value of the loudness. In that case, the algorithm increases the pulse rate and decreases the loudness. This process is repeated for the given number of generations. In general, the loudness decreases once a new best solution is found, while the rate of pulse emission decreases. For simplicity, the following values are commonly used: $\mathcal{A}_0 = 1$ and $\mathcal{A}_{min} = 0$, assuming that this latter value means that a bat has found the prey and temporarily stop emitting any sound. The evolution rules for loudness and pulse rate are as: $\mathcal{A}_i^{g+1} = \alpha \mathcal{A}_i^g$ and $r_i^{g+1} = r_i^0[1 - exp(-\gamma g)]$ where α and γ are constants. Note that for any $0 < \alpha < 1$ and any $\gamma > 0$ we have: $\mathcal{A}_i^g \to 0$, $r_i^g \to r_i^0$ as $g \to \infty$. Generally, each bat should have different values for loudness and pulse emission rate, which can be achieved by randomization. To this aim, we can take an initial loudness $\mathcal{A}_i^0 \in (0, 2)$ while the initial emission rate r_i^0 can be any value in the interval $[0, 1]$. Loudness and emission rates will be updated only if the new solutions are improved, an indication that the bats are moving towards the optimal solution.

4 The Proposed Method

For data fitting we consider a *free-form parametric Bézier curve* $\mathbf{\Phi}(\tau)$ *of degree* η, defined as:

$$\mathbf{\Phi}(\tau) = \sum_{j=0}^{\eta} \mathbf{\Delta}_j \phi_j^{\eta}(\tau) \tag{4}$$

where $\mathbf{\Delta}_j$ are vector coefficients called *control points*, $\phi_j^{\eta}(\tau)$ are the *Bernstein polynomials of index* j *and degree* η, given by:

$$\phi_j^{\eta}(\tau) = \binom{\eta}{j} \tau^j \ (1 - \tau)^{\eta - j}$$

and τ is the *curve parameter*, defined on a finite interval $[0, 1]$. Note that in this paper vectors are denoted in bold. By convention, $0! = 1$.

Suppose that we are provided with a list of data points $\{\mathbf{\Theta}_\mu\}_{\mu=\mathbf{1},\ldots,\chi}$ in $\mathbb{R}^2$ selected by a trained dermatologist from dermoscopy images. This list $\{\mathbf{\Theta}_\mu\}$ is always sorted and then joined to define a border between a skin lesion and the skin background, thus enclosing a region that contains the full skin lesion under analysis. Our goal is to compute the curve $\mathbf{\Phi}(\tau)$ approximating the data points better in the least-squares sense. This approximation scheme is particularly adequate to describe the border through a smooth mathematical equation instead of a (possibly large) list of data points connected by straight lines. To do so, we

have to minimize the least-squares error, Υ, defined as the sum of squares of the residuals:

$$\Upsilon = \sum_{\mu=1}^{\chi} \left(\mathbf{\Theta}_\mu - \sum_{j=0}^{\eta} \mathbf{\Delta}_j \phi_j^\eta(\tau_\mu) \right)^2 \tag{5}$$

where we need a parameter value τ_μ to be associated with each data point $\mathbf{\Theta}_\mu$, $\mu = 1, \ldots, \chi$. Considering the column vectors $\mathbf{\Phi}_j = (\phi_j^\eta(\tau_1), \ldots, \phi_j^\eta(\tau_\chi))^T$, $j = 0, \ldots, \eta$, where $(.)^T$ means transposition, and $\bar{\mathbf{\Theta}} = (\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_\chi)$, Eq. (5) becomes the following system of equations (called the *normal equation*):

$$\begin{pmatrix} \mathbf{\Phi}_0^T.\mathbf{\Phi}_0 & \ldots & \mathbf{\Phi}_\eta^T.\mathbf{\Phi}_0 \\ \vdots & \vdots & \vdots \\ \mathbf{\Phi}_0^T.\mathbf{\Phi}_\eta & \ldots & \mathbf{\Phi}_\eta^T.\mathbf{\Phi}_\eta \end{pmatrix} \begin{pmatrix} \mathbf{\Delta}_0 \\ \vdots \\ \mathbf{\Delta}_\eta \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{\Theta}}.\mathbf{\Phi}_0 \\ \vdots \\ \bar{\mathbf{\Theta}}.\mathbf{\Phi}_\eta \end{pmatrix} \tag{6}$$

which can be compacted as:

$$\mathcal{M}\,\mathcal{D} = \mathcal{R} \tag{7}$$

with $\mathcal{M} = \left[\sum_{j=1}^{\chi} \phi_l^\eta(\tau_j) \phi_i^\eta(\tau_j) \right]$, $\mathcal{R} = \left[\sum_{j=1}^{\chi} \mathbf{\Theta}_j \phi_l^\eta(\tau_j) \right]$ for $i, l = 0, \ldots, \eta$. If values are assigned to the τ_i, Eq. (7) is a classical linear least-squares minimization that can readily be solved by standard numerical techniques. But if the τ_i are unknowns, the problem becomes more difficult. Since the blending functions $\phi_j^\eta(\tau)$ are nonlinear in τ, 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.

We solve this general problem by applying the bat algorithm described above to determine suitable parameter values for the least-squares minimization of functional Υ according to (5). To this aim, each bat (potential solution) is a parametric vector $\mathcal{T}^k = (\tau_1^k, \tau_2^k, \ldots, \tau_\chi^k) \in [0,1]^\chi$, where the $\{\tau_\mu^k\}_{\mu=1,\ldots,\chi}$ are strictly increasing parameters. These parametric vectors are initialized with random values and then sorted. Application of our method yields new bats representing the potential solutions of this optimization problem. The process is performed iteratively for a given number of generations $\mathcal{G}_{max}$, until the convergence of the minimization of the error is eventually achieved. The bat with the best global value for our fitness function is taken as the final solution of our problem.

A critical issue for bat algorithm is the parameter tuning, which is well-known to be problem-dependent. Our choice has been fully empirical. We set the population size to 100 bats, as larger values increase the computational time without any significant improvement of the fitness function. The initial and minimum loudness, f_{max}, and parameter α are set to 0.5, 0, 2, and 0.6, respectively. We also set the initial pulse rate and parameter γ to 0.5 and 0.4, respectively. However, our results do not change significantly when varying these values slightly. All executions are performed for a total of 10,000 iterations.

Fig. 1. Dermoscopy images of skin lesions used in this paper.

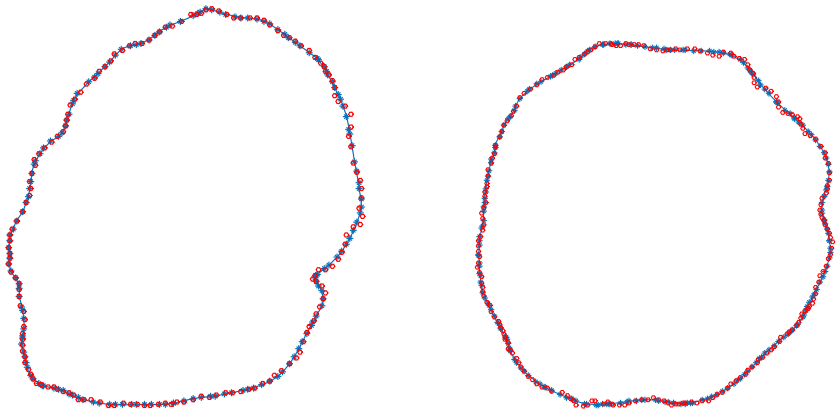

Fig. 2. Automatic fitting of the feature points for border detection for the examples in Figure 1: feature points (red empty circles), best fitting Bézier curve (solid blue line) and reconstructed feature points (blue stars).

5 Experimental Results

Our method has been applied to several examples of skin lesion images obtained from a digital image archive of the University Medical Center of Groningen, The Netherlands. In this paper we analyze only two of them because of limitations of space. They are displayed in Figure 1 and correspond to two dermoscopy images from which a collection of 162 and 220 feature points respectively have been marked by a specialist. They are displayed as red empty circles in Figure

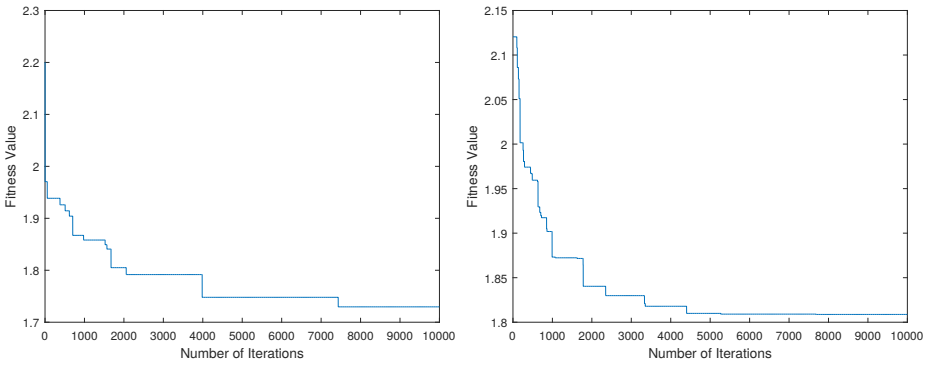

Fig. 3. Convergence diagrams for the examples in Figure 2.

2 for both cases. We applied our method to these examples by using Bézier curves of different degrees, and then selecting those minimizing the least-squares functional in Eq. (5). The best fitting Bézier curves obtained by our method, corresponding to $n = 55$ and $n = 40$ respectively, are displayed as a blue solid line in Fig. 2 left and right, respectively. The figure also displays the feature points reconstructed by our method as blue stars. As the reader can see, the method obtains a very good fitting of the data points. This observation is confirmed by our numerical results, where we obtain a value of $\Upsilon = 1.72959$ for the first example and $\Upsilon = 1.80875$ for the second one. We also computed the RMSE (root-mean square error), given by: $RMSE = \sqrt{\Upsilon/\chi}$ and obtained a value of 1.0321×10^{-1} and 9.0672×10^{-2}, respectively. We also noticed that the approximation is not optimal yet, as expected from an approximation method. In particular, the original data seems to be slightly more oscillating than the reconstructed curve in both cases. We remark, however, that a perfect matching between the original and the reconstructed features points is not actually required for clinical practice; the accuracy of our results (even if not totally optimal) is generally considered adequate for early diagnosis purposes. Figure 3 shows the convergence diagram of our method (i.e. the fitness function value Υ vs. the number of iterations) for both examples. All the computational work in this paper has been performed on a personal PC with a 2.6 GHz. Intel Core i7 processor and 8 GB. of RAM. The source code has been implemented by the authors in the programming language of the popular numerical program *Matlab*, version 2015b.

The present method can be improved in some ways. We plan to extend this approach by using piecewise polynomial schemes for better accuracy. We also plan to include some recent methods based on fusion thresholding to obtain the initial feature points without (or with minimal) human intervention to the aim to perform image segmentation in a more automatic way. We also plan to apply this technique to other interesting problems in other fields of medical imaging, not only dermoscopy images.

Acknowledgments

This work is supported by the project PDE-GIR of the European Union's Horizon 2020 research and innovation programme, Marie Sklodowska-Curie grant agreement No 778035, the Spanish Ministry of Economy and Competitiveness (Computer Science National Program) under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE), the project #JU12 of SODERCAN and European Funds FEDER (SODERCAN/FEDER UE), and the project EMAITEK of the Basque Government.

References

1. 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).
2. Argenziano, G., Soyer, H.P., De Giorgi, V.: *Dermoscopy: A Tutorial*. EDRA Medical Publishing & New Media (2002).
3. Barhak, J., Fischer, A.: Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. *IEEE Trans. on Visualization and Computer Graphics*, **7**(1) 1-16 (2001).
4. 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).
5. Dierckx, P.: *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford (1993).
6. Engelbretch, A.P.: *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons, Chichester, England (2005).
7. Gálvez, A., Iglesias A.: Efficient particle swarm optimization approach for data fitting with free knot B-splines. *Computer-Aided Design*, **43**(12) 1683-1692 (2011).
8. Gálvez, A., Iglesias, A.: Firefly algorithm for explicit B-Spline curve fitting to data points. *Mathematical Problems in Engineering*, (2013) Article ID 528215, 12 pages.
9. Gálvez A., Iglesias A.: From nonlinear optimization to convex optimization through firefly algorithm and indirect approach with applications to CAD/CAM. *The Scientific World Journal*, (2013) Article ID 283919, 10 pages.
10. Gálvez A., Iglesias A.: New memetic self-adaptive firefly algorithm for continuous optimization. *International Journal of Bio-Inspired Computation*, **8**(5), 300–317 (2016).
11. Gálvez A., Iglesias A., Avila, A., Otero, C., Arias, R., Manchado, C.: Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting. *Applied Soft Computing*, **26**, 90-106 (2015).
12. Gálvez, A., Iglesias, A., Cobo, A., Puig-Pey, J., Espinola, J.: Bézier curve and surface fitting of 3D point clouds through genetic algorithms, functional networks and least-squares approximation. *Lectures Notes in Computer Science*, **4706**, 680-693 (2007).
13. 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).
14. Gu, P., Yan, X.: Neural network approach to the reconstruction of free-form surfaces for reverse engineering. *Computer-Aided Design* **27**(1), 59-64 (1995).

15. Hoffmann, M.: Numerical control of Kohonen neural network for scattered data approximation. *Numerical Algorithms*, **39**, 175-186 (2005).
16. Iglesias, A., Echevarría, G., Gálvez, A.: Functional networks for B-spline surface reconstruction. *Future Generation Computer Systems*, **20**(8), 1337-1353 (2004).
17. Jing, L., Sun, L.: Fitting B-spline curves by least squares support vector machines. In: *Proc. of the 2nd. Int. Conf. on Neural Networks & Brain*. Beijing (China). IEEE Press (2005) 905-909.
18. Jupp, D.L.B.: Approximation to data by splines with free knots. *SIAM Journal of Numerical Analysis*, **15**, 328-343 (1978).
19. Kennedy, J., Eberhart, R.C., Shi, Y.: *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco (2001).
20. Knopf, G.K., Kofman, J.: Adaptive reconstruction of free-form surfaces using Bernstein basis function networks. *Engineering Applications of Artificial Intelligence*, **14**(5), 577-588 (2001).
21. 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).
22. 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).
23. Park, H.: An error-bounded approximate method for representing planar curves in B-splines. *Computer Aided Geometric Design* **21**, 479-497 (2004).
24. Park, H., Lee, J.H.: B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design* **39**, 439-451 (2007).
25. Schmid, P.: Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Transactions on Medical Imaging*, **18**(2), 164–171 (1999).
26. Ulker, E., Arslan, A.: Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Information Sciences*, **179**, 1483-1494 (2009).
27. Wang, W.P., Pottmann, H., Liu, Y.: Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics*, **25**(2), 214-238 (2006).
28. Yang, X.-S.: *Nature-Inspired Metaheuristic Algorithms (2nd. Edition)*. Luniver Press, Frome, UK (2010).
29. Yang, X.S.: A new metaheuristic bat-inspired algorithm. *Studies in Computational Intelligence*, **284**, 65–74 (2010).
30. Yang, X. S.: Bat algorithm for multiobjective optimization. *Int. J. Bio-Inspired Computation*, **3**(5), 267–274 (2011).
31. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, **29**(5), 464–483 (2012).
32. Yang, X.S.: Bat algorithm: literature review and applications. *Int. J. Bio-Inspired Computation*, **5**(3), 141–149 (2013).
33. Yoshimoto, F., Moriyama, M., Harada, T.: Automatic knot adjustment by a genetic algorithm for data fitting with a spline. *Proc. of Shape Modeling International'99*, IEEE Computer Society Press, 162-169 (1999).
34. Yoshimoto, F., Harada T., Yoshimoto, Y.: Data fitting with a spline using a real-coded algorithm. *Computer-Aided Design*, **35**, 751-760 (2003).
35. 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).