This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Immunological Approach for Full NURBS Reconstruction of Outline Curves from Noisy Data Points in Medical Imaging

Andrés Iglesias, Akemi Gálvez, and Andreina Avila

Abstract—Curve reconstruction from data points is an important issue for advanced medical imaging techniques, such as computer tomography (CT) and magnetic resonance imaging (MRI). The most powerful fitting functions for this purpose are the NURBS (non-uniform rational B-splines). Solving the general reconstruction problem with NURBS requires to compute all free variables of the problem (data parameters, breakpoints, control points and their weights). This leads to a very difficult non-convex, nonlinear, high-dimensional, multimodal, continuous optimization problem. Previous methods simplify the problem by guessing the values for some variables and computing only the remaining ones. As a result, unavoidable approximations errors are introduced. In this paper, we describe the first method in the literature to solve the full NURBS curve reconstruction problem in all its generality. Our method is based on a combination of two techniques: an immunological approach to perform data parameterization, breakpoint placement, and weight calculation, and least squares minimization to compute the control points. This procedure is repeated iteratively (until no further improvement is achieved) for higher accuracy. The method has been applied to reconstruct some outline curves from MRI brain images with satisfactory results. Comparative work shows that our method outperforms the previous related approaches in the literature for all instances in our benchmark.

Index Terms—Medical imaging, curve reconstruction, NURBS, artificial immune systems, metaheuristics, optimization.

1 Introduction

1.1 Curve and Surface Reconstruction

Fitting data points to curves and surfaces (also known as *curve* and *surface reconstruction*, respectively) is a very important research field with outstanding applications in many industrial and applied domains. Typical examples include geometric modeling and processing [2], [15], and reverse engineering for computer-aided design and manufacturing (CAD/CAM) [16], [17], [37], [49]. Medical sciences and bioinformatics are two important areas where curve and surface reconstruction have also had a major impact. A number of powerful techniques and devices developed during the last few decades have made it possible to create realistic and accurate visual representations of the interior of our bodies for clinical diagnosis and treatment in either noninvasive or minimally invasive way. Medical imaging techniques extremely common (even pervasive) nowadays – such as computer tomography (TC) and magnetic resonance imaging (MRI) – have taken advantage from the spectacular advances in digital geometry processing to generate three-dimensional images of inner organs and other parts of the human body for diagnostic and therapeutic purposes.

These advances prompted a sudden boom in the use of advanced medical imaging during the first decade of this century. Between 1993 and 2006, the annual unit sales of CT and MRI units nearly tripled [36], and CT grew at an impressive annual rate of 14.3 percent during the period 2000-2005. Very recently, the world's largest medical imaging study made headlines in the newspapers [45]: about 100,000 British volunteers (from half a million middle age people already taking part in *UKBioBank*, a massive project set up in 2006 to gather medical and lifestyle data on the UK population) will have their hearts, brains, bones and body fat scanned to help doctors find links between diseases [45]. Reasons for this popularity of medical imaging are varied, but several studies linked its use to various positive outcomes, such as *"longer life expectancy, declines in mortality, less need for exploratory surgery, fewer hospital admissions, and shorter lengths of hospital stays"* [36].

Curve and surface reconstruction are at the core of digital medical imaging techniques. Think about - for instance - the multiplanar CT, which consists of generating a volumetric shape from a collection of axial slices. Typical solutions to this problem are based on surface reconstruction either from a series of planar curves defining the boundary of that volume at different slices or from a collection of three-dimensional data points captured at different layers. Modern CT/MRI machines can provide images in classical DICOM (Digital Imaging and Communications in Medicine) file format with a spacing between consecutive slices of 1 mm. or even less, thus providing more detailed anatomical

- *A. Iglesias is with the Department of Information Science, Faculty of Sciences, Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan, and the Department of Applied Mathematics and Computational Sciences, E.T.S.I. Caminos, Canales y Puertos, University of Cantabria, Avda. de los Castros, s/n, 39005, Santander, Spain. E-mail: iglesias@unican.es.*
- *A. Gálvez is with the Department of Information Science, Faculty of Sciences, Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan, and the Department of Applied Mathematics and Computational Sciences, E.T.S.I. Caminos, Canales y Puertos, University of Cantabria, Avda. de los Castros, s/n, 39005, Santander, Spain. E-mail: galveza@unican.es.*
- *A. Avila is with the Department of Applied Mathematics and Computational Sciences, E.T.S.I. Caminos, Canales y Puertos, University of Cantabria, Avda. de los Castros, s/n, 39005, Santander, Spain. E-mail: andreina.avila@alumnos.unican.es.*

Manuscript received 30 Sep. 2016; revised 11 Feb. 2017;

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

data for reconstruction. Other medical imaging techniques make also use of intensive curve (resp., surface) reconstruction algorithms to derive boundary curves (resp., surfaces) between surfaces (resp., volumes) corresponding to different organs or tissues.

Other important field of application of curve and surface reconstruction algorithms is biomedical engineering, particularly for the generation of prosthesis and customized medical implants. The widespread popularization and sharp decline of prices of 3D printers, additive layer manufacturing technologies, and handheld scanning devices (that have gone from very expensive to easily affordable in a matter of few years) have transformed the typical landscape of surgical implants and prosthesis. Researchers are now using specialized 3D printers to grow human tissues and even organs. Intensive research is currently conducted on new biomimetic materials such as polymers to help the body rebuild damaged tissue on printed scaffolds. Also, bio-printing is making significant contributions in helping patients to fully understand their health condition and enhancing the comprehension of health problems such as tumors in inner organs from their 3D printed counterpart models.

1.2 NURBS in Medical Imaging

Several approaches - such as polygonal meshes and CSG (Constructive Solid Geometry) models - have been proposed for shape reconstruction. However, the most reliable and accurate reconstruction techniques are based on mathematical curves and surfaces, usually described in parametric form. Among them, the so-called free-form parametric entities (such as Bézier, B-splines and NURBS) have become very popular in computer graphics and many related fields. Arguably, the most powerful ones are the *NURBS*, an acronym for *N*on-*U*niform *R*ational *B*-*S*plines. They are an extension of the classical piecewise polynomial B-splines, so they share many useful properties. However, the NURBS have more degrees of freedom and, therefore, the ability to represent a larger class of shapes than the polynomial B-splines (e.g., conics and quadrics).

Owing to these valuable features, NURBS have become an essential tool for several medical imaging purposes. For instance, they offer an attractive alternative to uniform polynomial B-splines in modeling the displacement vector field (DVF) in deformable image registration. The NURBS capability to add extra flexibility through breakpoint placement and weights provides new degrees of freedom and increases local control as well as the ability to explicitly represent topological discontinuities [31]. The NURBS are also widely used in the generation of computational human phantoms (models of the human body that can replicate the internal structure and organs of the human body very accurately). Because the anatomy and physiology of the phantom are well known, computer-generated phantoms are very useful for the development and assessment of medical imaging devices, image processing techniques, and volumetric reconstruction algorithms. While the first generations of digital phantoms were mostly based on either simple primitives (stylized phantoms) or volumetric pixels (voxel phantoms), current models are increasingly using NURBS for more realistic representation and higher accuracy. In fact, the NURBS allow for 4D phantoms, where simulations are typically performed not only in the three dimensional space but also over the time. An illustrative example is the 4D XCAT phantom from Johns Hopkins Medicine, where NURBS are used to construct organ shapes using the three-dimensional Visible Human CT dataset as a preliminary basis, and then extended to 4D using 4D tagged MRI and CT data for cardiac and respiratory motions, respectively. This technology also allows to take advantage of the last developments in genetics and molecular biology regarding small animal imaging, an emerging area in which new devices and techniques for molecular imaging research are needed. An illustrative example is the 4D MOBY phantom, where dynamic NURBS surfaces defined by series of temporal curves are applied to generate a continuous model.

Yet, the most important application of NURBS in medical imaging comes from curve and surface reconstruction. Typically this task involves a first segmentation stage, where the massive cloud of data points is partitioned into different regions according to a threshold value; then, a shape reconstruction stage is carried out by using individual NURBS for each region, which are subsequently combined together to construct a single full model. At its turn, the shape reconstruction stage can be further split into several subtasks: data parameterization, breakpoint placement, weight calculation, and control point computation (see Section 5.1 for details).

Shape reconstruction with NURBS is extremely difficult, because it requires to compute many different sets of unknowns (data parameters, breakpoints, control points, and weights) that are strongly intertwined in a highly complicated and nonlinear way. As a consequence, the problem of computing all these free variables cannot be decomposed into independent sub-problems for each particular set of variables, leading to a challenging optimization problem. The problem has become so difficult that all classical mathematical optimization techniques have failed so far to provide a satisfactory solution for the general case.

A classical way to overcome this limitation has been to simplify the general problem by manually setting the values for some of those sets of variables, then perform optimization for the remaining ones. Obviously, the resulting reconstructed shape is not fully optimized and contains errors derived from the initial choice of values for some unknowns. Moreover, making a good initial guess of suitable values requires considerable expertise. This task is also very time-consuming because it is performed on a trial-and-error basis and is strongly problem-dependent. Clearly, there is a need for powerful methods to compute all free variables *simultaneously and automatically*. This is one of the main contributions of this paper.

1.3 Aims and Structure of this Paper

This paper introduces a new method for full NURBS reconstruction of outline curves in medical imaging from a given set of noisy data points. This problem is very challenging, as it becomes a difficult non-convex, nonlinear, high-dimensional, multimodal, continuous optimization problem. The differential feature of this method is its

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

generality: although there are some previous methods using NURBS curves, they are strongly limited because they do not compute all free variables of the problem (data parameters, breakpoints, control points and their weights). Instead, they simplify the problem by making an initial choice of values for some of those variables and just computing the remaining ones. Obviously, unavoidable approximations errors are introduced during this simplification process. On the contrary, our method uses totally general NURBS curves as fitting functions; therefore, *all free variables of the problem are fully computed*. To the best of our knowledge, this is *the first method solving the problem of curve reconstruction with NURBS in all its generality*.

Our method is based on a combination of two techniques: an immunological approach - called modified clonal selection algorithm - to perform data parameterization, breakpoint placement, and weight calculation, and least squares minimization to compute the control points. This procedure is repeated iteratively (until no further improvement is achieved) for higher accuracy.

The structure of this paper is as follows: Section 2 describes the previous work in the field. The mathematical background about B-spline and NURBS curves required to understand the paper along with the problem of curve reconstruction with NURBS are explained in Section 3. Then, Section 4 describes briefly the artificial immune systems, with emphasis on the clonal selection algorithm, the immunological approach used in this paper. Our proposed method is described in detail in Section 5, including the important issues of parameter tuning and computational complexity. Section 6 describes our experimental results, some implementation issues, and the computational execution time. Comparative work of our method with other previous related approaches is reported in Section 7. The paper closes with the main conclusions of this work and some ideas for future work.

2 Previous Work

Shape reconstruction with free-form parametric curves has been an active field of research for many years. First approaches in the field were numerical procedures based on classical mathematical optimization methods [10], [11]. These early works focused on the parameterization step, i.e., how to obtain good parametric values for the data points. At that time, some (currently common) techniques such as the uniform, arc-length, and centripetal parameterizations were developed [11], [42]. The uniform parameterization (where all data parameters are equally spaced) is a popular choice because of its simplicity. However, in many practical settings it is more convenient to obtain a parameterization as close as possible to the arc-length parameterization. With this parameterization, constant steps on the parametric domain automatically transforms into constant distances along the arc-length parameterized curve. This property has many useful applications, such as measuring distances along the curve. The centripetal method is an extension of the arc-length parameterization that yields better results for shapes exhibiting sharp turns [42].

Once the parameterization step is solved, we have still to compute the breakpoints. Early approaches proceeded by setting the number of breakpoints *a priori* and then computing their location according to some formula [32], [40], [41], [50]. The simplest way to do it is to consider equally spaced values, the so-called *uniform breakpoint vector*. Very often, this approach is not appropriate as it may lead to singular systems of equations and does not reflect the real distribution of data. A more refined procedure consists of the *averaging method* and its variations [42], which allocate breakpoints to ensure that every interval span contains at least one parametric value.

Years later, it was shown that B-spline curve reconstruction improves dramatically if the breakpoints are treated as free variables of the problem [33]. Traditional techniques change the number of breakpoints through breakpoint insertion or removal. Usually, these methods require terms or parameters (e.g., tolerance errors, smoothing factors, or cost functions) whose values are chosen in subjective fashion [10], [24], [33]. Therefore, they fail to *automatically* generate a good breakpoint vector. In addition, these methods are usually be prone to errors and time-consuming. Other approaches use error bounds [40], curvature-based squared distance minimization [50], or dominant points [41]. In general, they perform well but require some strong constraints (such as high differentiability, closed curves, noiseless data) which are very difficult to meet for many real-world applications. Other methods use curvature information extracted from input data [34], [35]. They are, therefore, restricted to smooth data points and are very sensitive to noise in data.

In the last few years, some artificial intelligence techniques have been applied to this problem. Most of these methods rely on some kind of neural networks, such as standard neural networks [26], or Kohonen's self-organizing maps nets [28], sometimes combined with partial differential equations [1] or other approaches [30]. The generalization of these methods to functional networks (where the scalar weights are replaced by functions) is analyzed in [29]. The application of support vector machines to this problem is reported in [32]. All these methods are restricted to particular (usually very simple) cases and do not address the general case.

A promising line of research is given by the *nature-inspired metaheuristic algorithms*, which have become very popular to solve difficult optimization problems [52]. Basically, they refer to a broad set of computational techniques of different types that either imitate or are simply inspired by processes typically found in nature [51]. Among them, the majority are based on successful characteristics of biological systems; such methods are labeled as *bio-inspired methods* as well. The interested reader is kindly referred to the nice book [51] for an introduction to the field of nature-inspired metaheuristic algorithms; see also [12] for a brief review of nature-inspired algorithms for optimization. The nature-inspired metaheuristics applied to B-spline curves can be classified into two groups: discrete approaches and continuous approaches. Methods in the former group convert the original continuous problem into a discrete combinatorial optimization problem solved by either genetic algorithms [43], [53] or artificial immune systems [47]. As expected, this conversion process introduces large discretization errors, making them both inaccurate and unreliable for real-world problems. The continuous methods avoid the discretization

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

errors but they generally restricted to the simpler case of explicit functions (that do not require data parameterization) and cannot deal with the parametric case [14], [18], [21], [23], [54]. The hybrid approaches tend to be slow, time-consuming and difficult to implement, as they usually require additional sophisticated methods [17], [22]. The methods in [19], [20], [55] consider the case of parametric B-spline curves but only compute some of the variables, either the data parameters [19] or the breakpoints [20], [55], so they cannot solve the general case. In addition, the work in [55] is strongly limited to closed curves. Finally, none of these methods considers the addition of weights and are therefore, limited to the polynomial case. As a consequence, they cannot deal with the (more difficult) case of NURBS curves.

To summarize, there is no method reported in the literature so far to solve the general curve reconstruction problem with NURBS curves. The method presented in this paper is aimed at filling this gap.

3 Mathematical Background

3.1 B-spline and NURBS Curves

3.1.1 B-spline Basis Functions

Let $\boldsymbol{\Upsilon} = \{\mu_0 = \alpha, \mu_1, \ldots, \mu_\sigma, \mu_{\sigma+1} = \beta\}$ be a non-decreasing sequence of non-negative real numbers on the compact interval $[\alpha, \beta] \subset \mathbb{R}$. Note that in this paper vectors are denoted in bold. Without loss of generality, $[\alpha, \beta]$ can be assumed to be $[0, 1]$. The elements of $\boldsymbol{\Upsilon}$ are called *breakpoints*. For each sequence $\boldsymbol{\Upsilon}$, the *k-th B-spline basis function* $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ *of order* ρ (or, equivalently, degree $\rho - 1$) *and breakpoints* $\boldsymbol{\Upsilon}$ is defined by the Cox-de Boor recurrence relations [3], [11], [42]:

$$\varphi_{k,1}(\tau, \boldsymbol{\Upsilon}) = \begin{cases} 1 & \text{if } \mu_k \leq \tau < \mu_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $k = 0, \ldots, \sigma$, and:

$$\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon}) = \frac{\tau - \mu_k}{\mu_{k+\rho-1} - \mu_k} \varphi_{k,\rho-1}(\tau, \boldsymbol{\Upsilon}) + \frac{\mu_{k+\rho} - \tau}{\mu_{k+\rho} - \mu_{k+1}} \varphi_{k+1,\rho-1}(\tau, \boldsymbol{\Upsilon}) \tag{2}$$

for $\rho > 1$ and $k = 0, \ldots, \sigma - \rho + 1$. If necessary, the convention $\frac{0}{0} = 0$ in Eq. (2) is applied.

It is worthwhile to mention that, according to Eq. (2), the $\{\mu_j\}_j$ are not only in the numerator but also in the denominator of $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$. This means that $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ is a nonlinear function of $\boldsymbol{\Upsilon}$. Note that k-th B-spline basis function of order 1, $\varphi_{k,1}(\tau, \boldsymbol{\Upsilon})$, is a piecewise constant function with value 1 on the interval $[\mu_k, \mu_{k+1})$, called the *support* of $\varphi_{k,1}(\tau, \boldsymbol{\Upsilon})$, and zero elsewhere.

From Eq. (2) it is clear that every basis function $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ of order $\rho > 1$ is a combination of two consecutive basis functions of order $\rho - 1$, so its support is given by the interval $[\mu_k, \mu_{k+\rho})$. Also, for $\rho > 1$, the functions $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ fulfill the following two conditions:

(C1) $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ is a polynomial spline of degree up to $\rho - 1$ on each interval $[\mu_k, \mu_{k+1})$, and

(C2) $\varphi_{k,\rho}(\tau, \boldsymbol{\Upsilon})$ and its derivatives up to order $\rho - 1$ are continuous on $[\mu_k, \mu_{k+1})$.

The dimension of the vector space of functions satisfying conditions (C1) and (C2) is $\sigma + \rho$. The vector of breakpoints $\boldsymbol{\Upsilon}$ yields $\sigma - \rho + 2$ linearly independent basis functions of order ρ. The remaining $2(\rho - 1)$ basis functions are obtained by imposing the boundary conditions $\mu_{-\rho+1} = \mu_{-\rho+2} = \cdots = \mu_{-2} = \mu_{-1} = \mu_0 = \alpha$ and $\mu_{\sigma+1} = \mu_{\sigma+2} = \cdots = \mu_{\sigma+\rho} = \beta$. With this choice of boundary breakpoints all basis functions vanish outside the interval domain $[\alpha, \beta]$.

3.1.2 B-spline Curves

A *B-spline curve* $\boldsymbol{\Phi}(\tau, \boldsymbol{\Upsilon})$ *of order* ρ *and breakpoints* $\boldsymbol{\Upsilon}$ is a piecewise polynomial curve given by:

$$\boldsymbol{\Phi}(\tau, \boldsymbol{\Upsilon}) = \sum_{i=0}^{\delta} \boldsymbol{\Xi}_i \, \varphi_{i,\rho}(\tau, \boldsymbol{\Upsilon}) \tag{3}$$

where $\boldsymbol{\Xi} = \{\boldsymbol{\Xi}_i\}_{i=0,\ldots,\delta}$ are vector coefficients of the curve in $\mathbb{R}^d$ (typically, $d = 2$ or $d = 3$) called *control points* and $\{\varphi_{i,\rho}(\tau, \boldsymbol{\Upsilon})\}_i$ are the basis functions defined according to Eqs. (1)-(2). B-spline curves are very popular in many fields because of their remarkable properties, such as:

- *Invariance under affine transformations*: any affine transformation (e.g. scaling, rotation, or translation) on the curve can be done directly on its control points.
- *Local support*: a single span of a B-spline curve is controlled by only ρ control points, and any control point only affects ρ spans. In particular, any displacement of the control point $\boldsymbol{\Xi}_k$ in Eq. (3) only affects the interval $[\mu_k, \mu_{k+\rho})$ while other parts of the curve remain unchanged. This is a very convenient feature because it means that the curve can be modified locally (i.e., the curve has the desirable property of *local control*, as opposed to global control schemes such as the Bézier curves). Note that the region affected by the control point displacement can be controlled by moving the breakpoints $\mu_k, \mu_{k+1}, \ldots, \mu_{k+\rho}$.
- *Geometric continuity*: This property comes directly from the previous one. Each piece of the B-spline curve is a polynomial of class $C^{\rho-1}$. The curve is also continuous at the breakpoints. When all internal breakpoints are distinct, its derivatives are also continuous up to degree $\rho - 1$. If internal breakpoints are repeated λ times at a given value of τ, the order of continuity is reduced to $C^{\rho-\lambda}$ at that point.
- *Extension of Bézier curves*: the Bézier curves are included in the B-spline formalism. In particular, any Bézier curve of degree κ is a B-spline curve with no internal breakpoints and both end breakpoints repeated $\kappa + 1$ times.

3.1.3 NURBS Curves

The polynomial representation in Eq. (3) is not powerful enough to represent a variety of shapes, particularly the conics (e.g., circles, ellipses, and hyperbolas). One way to overcome this limitation is to use homogeneous coordinates (see [11], [42] for details). The basic idea is to consider the projection of the standard polynomial B-spline curve

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

in $\mathbb{R}^{d+1}$, with new control points $\boldsymbol{\Xi}_i^h$. The resulting curve projected onto $\mathbb{R}^d$ is called a NURBS curve. Mathematically, a *NURBS curve* $\boldsymbol{\Psi}(\tau, \boldsymbol{\Upsilon})$ *of order* ρ *and breakpoints* $\boldsymbol{\Upsilon}$ can be described as a quotient of two polynomials:

$$\boldsymbol{\Psi}(\tau, \boldsymbol{\Upsilon}) = \frac{\sum_{i=0}^{\delta} \omega_i \, \boldsymbol{\Xi}_i \, \varphi_{i,\rho}(\tau, \boldsymbol{\Upsilon})}{\sum_{i=0}^{\delta} \omega_i \, \varphi_{i,\rho}(\tau, \boldsymbol{\Upsilon})} \tag{4}$$

where ω_j is the last coordinate of the homogeneous control point $\boldsymbol{\Xi}_i^h$. This set of new scalar parameters $\{\omega_i\}_{i=0,\ldots,\delta}$, called *weights*, provides us with additional degrees of freedom for better shape approximation. They also increase the model complexity, however, as we introduce a new set of parameters that have to be computed as well.

The boundary constraints imposed in Section 3.1.1 have the effect to force the curve to interpolate the first and last control points, i.e., $\boldsymbol{\Psi}(\alpha, \boldsymbol{\Upsilon}) = \boldsymbol{\Xi}_0$ and $\boldsymbol{\Psi}(\beta, \boldsymbol{\Upsilon}) = \boldsymbol{\Xi}_\delta$, a very desirable property in medical applications and many other fields. Note however that the method described in this paper is actually independent of these constraints.

NURBS curves are an extension of the B-spline curves. If all $\{\omega_j\}_j$ are identical, Eq. (4) automatically transforms into Eq. (3). As a consequence, the NURBS inherit many of the valuable B-spline properties, such as those discussed in Section 3.1.2. However, NURBS are even more powerful and have remarkable advantages over the polynomial B-splines. For instance, they are also invariant under projective transformations, and provide a common mathematical form for both standard analytical shapes (such as the conics, which are not properly represented with polynomial B-spline curves) and free-form shapes. This explains why NURBS are currently the standard *de facto* for CAD/CAM, computer graphics and many other fields. Because of these reasons, in this paper we will focus on the problem of curve reconstruction with NURBS curves, discussed in next section.

3.2 Curve Reconstruction with NURBS

Let us consider an input set of measured data points $\{\boldsymbol{\Theta}_k\}_{k=1,\ldots,\eta}$, obtained by computer tomography, magnetic resonance imaging, or any other medical imaging technique. The goal of curve reconstruction with NURBS consists of obtaining the NURBS curve $\boldsymbol{\Psi}(\tau, \boldsymbol{\Upsilon})$ of a given order ρ approximating better the data points $\{\boldsymbol{\Theta}_k\}_k$. Since the fitting curve is parametric, a suitable parameterization of data points is needed, so that each data point $\boldsymbol{\Theta}_k$ is associated with a data parameter value, denoted as $\zeta_k \in [\alpha, \beta]$. Due to the boundary conditions introduced in previous section, it is possible to take $\boldsymbol{\Psi}(\zeta_1, \boldsymbol{\Upsilon}) = \boldsymbol{\Theta}_1$ and $\boldsymbol{\Psi}(\zeta_\eta, \boldsymbol{\Upsilon}) = \boldsymbol{\Theta}_\eta$ and perform approximation on the remaining parameters:

$$\boldsymbol{\Theta}_k \approx \boldsymbol{\Psi}(\zeta_k, \boldsymbol{\Upsilon}) = \frac{\sum_{i=0}^{\delta} \omega_i \, \boldsymbol{\Xi}_i \, \varphi_{i,\rho}(\zeta_k, \boldsymbol{\Upsilon})}{\sum_{i=0}^{\delta} \omega_i \, \varphi_{i,\rho}(\zeta_k, \boldsymbol{\Upsilon})} \tag{5}$$

for $k = 2, \ldots, \eta-1$. Introducing the rational basis functions:

$$\xi_{j,\rho}(\tau, \boldsymbol{\Upsilon}) = \frac{\omega_j \, \varphi_{j,\rho}(\tau, \boldsymbol{\Upsilon})}{\sum_{i=0}^{\delta} \omega_i \, \varphi_{i,\rho}(\tau, \boldsymbol{\Upsilon})} \tag{6}$$

the Eq. (5) can be written as a linear combination of these rational basis functions, as:

$$\boldsymbol{\Theta}_k \approx \boldsymbol{\Psi}(\zeta_k, \boldsymbol{\Upsilon}) = \sum_{i=0}^{\delta} \boldsymbol{\Xi}_i \, \xi_{j,\rho}(\zeta_k, \boldsymbol{\Upsilon}) \tag{7}$$

Eq. (7) can be written in matrix notation as:

$$\boldsymbol{\Theta} = \boldsymbol{\Delta}.\boldsymbol{\Xi} \tag{8}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_{\eta-1})^T$, $\boldsymbol{\Xi} = (\boldsymbol{\Xi}_0, \ldots, \boldsymbol{\Xi}_\delta)^T$, $\boldsymbol{\Delta} = \left(\{\xi_{j,\rho}(\zeta_k, \boldsymbol{\Upsilon})\}_{j=0,\ldots,\delta \,;\, k=2,\ldots,\eta-1}\right)$ is the matrix of sampled B-spline rational basis functions, and $(.)^T$ represents the transpose of a vector or matrix. The dimension of the search space $D \subset \mathbb{R}^d$ in Eq. (8) is given by: $d(\delta+1)+\sigma+\eta+\delta-1$, which could be very large (up to thousands and even millions) for very complicated shapes.

System (8) is overdetermined, because it is commonly expected to approximate the data with far fewer parameters than the number of data points. Consequently, the matrix of basis functions $\boldsymbol{\Delta}$ is not invertible and no direct solution can be obtained. In this case, we consider the least squares approximation of (5), defined as the minimization problem given by:

$$\Lambda = \underset{\substack{\{\boldsymbol{\Xi}_i\}_i \\ \{\mu_j\}_j \\ \{\zeta_k\}_k \\ \{\omega_i\}_i}}{\text{minimize}} \left(\sum_{k=2}^{\eta-1} \left\| \boldsymbol{\Theta}_k - \frac{\sum_{i=0}^{\delta} \omega_i \, \boldsymbol{\Xi}_i \, \varphi_{i,\rho}(\zeta_k, \boldsymbol{\Upsilon})}{\sum_{i=0}^{\delta} \omega_i \, \varphi_{i,\rho}(\zeta_k, \boldsymbol{\Upsilon})} \right\|_{\ell_2} \right) \tag{9}$$

where ℓ_2 represents the Euclidean norm (although any other norm might be used instead).

In many real-world examples there is no information available about the problem other than the data points, meaning that all free variables of the problem:

- control points, $\{\boldsymbol{\Xi}_i\}_i$,
- data parameters, $\{\zeta_k\}_k$,
- breakpoints: $\{\mu_j\}_j$,
- and weights: $\{\omega_i\}_i$,

are to be calculated. This is a very difficult and challenging task. The main reason is that all these free variables are related to each other in a highly nonlinear and complicated way. In fact, the computation of any of these sets of variables requires to know in advance suitable values for the other sets, thereby preventing us from solving the general problem. This explains why previous methods in the literature are based on making a "reasonable" choice of values for some of those variables, which are subsequently used to compute the remaining values of the others (see our discussion about previous work in Section 2).

Although this procedure might be acceptable for some applications, it has several drawbacks: its performance is strongly dependent on the choice of initial values for the

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

fixed variables, a task that typically requires a lot of experience from the user. In addition, this choice is problem-dependent, meaning that good values for a particular example might not work well for another given example. As a consequence, this process remains mostly manual and no automatic procedure is usually applied. Moreover, the problem is non-convex, since the curve is a non-convex function of the breakpoints. Very often, it is also a multimodal problem, since there could be several (local and/or global) minima of the objective function.

To summarize, Eq. (9) represents a *strongly non-convex nonlinear high-dimensional multimodal continuous optimization problem*. The method proposed in this paper solves this (extremely difficult) problem by applying a powerful bio-inspired immunological approach called artificial immune systems, described in next section.

4 Immunological Approach for Optimization

4.1 Artificial Immune Systems

In recent years there is an increasing interest in biological systems as an inspiring source of ideas for new computational algorithms for optimization. Amongst them, the *Artificial Immune Systems* (AIS) have shown a remarkable ability to solve complex optimization problems in several fields. The term AIS refers to a powerful computational methodology mimicking the biological immune system of humans and other mammals. Therefore, rather than a single technique, AIS consists of a group of computationally intelligent systems inspired by the behavior and processes of the immune system. The reader is kindly referred to [4], [8], [9] for a gentle introduction to the field. See also [46] and [27] for further details about the theoretical advances on AIS and some areas of application, respectively.

Unfortunately, the immune system is far more complicated than it was anticipated. As a result, no single AIS encompasses all features of a real immune system. Instead, there are several models in AIS, each focused on the implementation of one or a few of those features. Relevant examples of AIS models include negative selection [13], artificial immune network [39], dendritic cells [25] and clonal selection [6]. Because of its appealing features regarding the optimization of multimodal functions [7] in this paper we focus on the clonal selection algorithm.

4.2 The Clonal Selection Algorithm

4.2.1 Main Concepts and Features

The *Clonal Selection Algorithm* (CSA) is an immunological approach proposed in 2000 by De Castro and Von Zuben [5], [6], [7]. It based on the clonal selection principle, a widely accepted theory that explains the basic features of an adaptive immune response to an antigenic stimulus. When we are exposed to an antigen, Ag, our immune system responds by producing *antibodies*, Ab's, which are molecules whose aim is to recognize and bind to Ag's. Each single type of Ab is relatively specific for the Ag. The degree of specificity is called the *affinity* of the couple Ag-Ab. According to the clonal selection theory, only those cells recognizing the antigens are selected to proliferate. Such cells are subject to an *affinity maturation* process, which improves their affinity to the selective Ag's over the time.

The learning process of the immune system involves raising the relative population size and affinity of the lymphocytes successful in recognizing a given Ag. A clone will be created temporarily and those progenies with low affinity will be discarded. Since an organism is expected to encounter a given Ag several times during its lifetime. the effectiveness of the immune response to secondary encounters is enhanced by the presence of memory cells associated with the first infection, capable of producing high-affinity Ab's just after several encounters.

Ab's in a memory response show a higher affinity than those of the early primary response. This process is called the *maturation* of the immune response. Then, random changes (*mutation*) are introduced into the genes responsible for the Ag-Ab interactions, increasing occasionally the affinity of the Ab. A rapid accumulation of mutations is necessary for a fast maturation of the immune response, even although most of the changes lead to poorer Ab's. When a B cell recognizes an antigen, it is stimulated to divide at an extremely high rate of somatic mutation, about five or six orders of magnitude greater than the normal rate of mutation. This process is called *somatic hypermutation*, and is regulated by the fact that cells with low-affinity receptors may be further mutated and die if they do not improve their clone size or antigenic affinity; on the contrary, hypermutation may become gradually inactive in cells with high-affinity Ab receptors.

Usually, the problem to be solved is represented through a Ag-Ab codification and we use a distance measure (called the *affinity measure*), to calculate their degree of interaction. This affinity, represented by $Af(Ab, Ag)$, can be measured as the distance between two arrays or vectors. If they are real-valued vectors, the Manhattan or Euclidian distances can be used; if they are binary symbols, the Hamming distance is usually applied (see [4] for more details).

4.2.2 The Algorithm

Fig. 1 shows the flowchart of the original CSA, designed for pattern recognition purposes [5], [7]. The algorithm considers two populations: a set of antigens $\mathbf{Ag}_{\{M\}}$ and a set of antibodies $\mathbf{Ab}_{\{N\}}$. For the sake of clarity, cardinality is indicated by the subindexes within brackets. The latter is further divided into two subsets: memory Ab set, $\mathbf{Ab}_{\{m\}}$, and remaining Ab set, $\mathbf{Ab}_{\{r\}}$, such that $m+r=N$. We also keep track of two other sets: the set $\mathbf{Ab}_{\{n\}}$ of the n Ab's with the highest affinities to a given Ag, and the set $\mathbf{Ab}_{\{d\}}$ of the d new Ab's that will replace the low-affinity Ab's from $\mathbf{Ab}_{\{r\}}$. The algorithm can be summarized as follows:

1) Random choice of an antigen $\mathbf{Ag}_j$. It is presented to all antibodies of $\mathbf{Ab}_{\{N\}}$.
2) Compute the vector affinity $\mathbf{F} = (F_1, F_2, \ldots, F_N)$ where $F_i = Af(\mathbf{Ab}_i, \mathbf{Ag}_j)$.
3) Select the n highest affinity components of $\mathbf{F}$ to generate $\mathbf{Ab}_{\{n\}}$.
4) Elements of $\mathbf{Ab}_{\{n\}}$ will be cloned adaptively. The number of clones is proportional to the affinity: the higher the affinity, the higher the number of clones.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

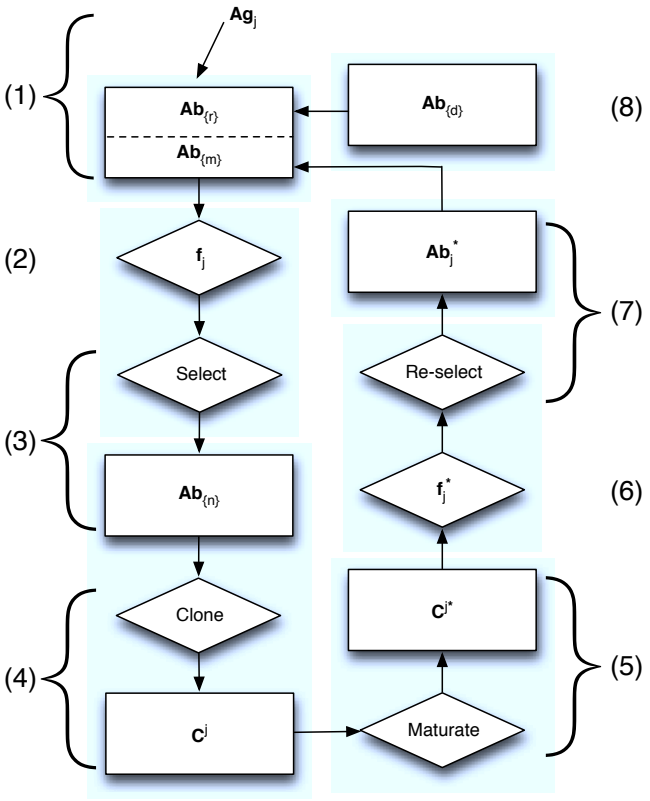

Fig. 1. Flow chart of the clonal selection algorithm described in [5], [7] for pattern recognition purposes.

Such amount is given by: $N_c = \sum_{h=1}^{n} round\left(\frac{\beta.N}{h}\right)$ where N_c represents the number of clones, β is a positive number that plays the role of a multiplying factor, N is the total number of Ab's and $round(.)$ is the operator that rounds its argument toward the closest integer.

5) The clones resulting from the previous step are subjected to somatic hypermutation. The affinity maturation rate is inversely proportional to the antigenic affinity: the higher the affinity, the smaller the maturation rate.
6) Compute the vector affinity of $\mathbf{Ag}_j$ with respect to the new matured clones.
7) From this set of matured clones, select the one with the highest affinity to be candidate to enter into the set $\mathbf{Ab}_{\{n\}}$. If $Af(\mathbf{Ab}_k, \mathbf{Ag}_j) > Af(\mathbf{Ab}_l, \mathbf{Ag}_j)$ for a given $\mathbf{Ab}_l \in \mathbf{Ab}_{\{n\}}$, then $\mathbf{Ab}_k$ will replace $\mathbf{Ab}_l$.
8) Replace the d Ab's with lowest affinity in $\mathbf{Ab}_{\{r\}}$ by new individuals in $\mathbf{Ab}_{\{r\}}$.

Each execution of the previous steps for all given $\mathbf{Ag}_j$, $(j = 1, \ldots, M)$ is called a *generation*. The algorithm is repeated for a certain number of generations, N_{gen}, a parameter that is problem-dependent. This algorithm is well-known to be very well suited for optimization problems [6], [7], [27].

4.2.3 Modified Algorithm for Multimodal Problems

The previous algorithm is only applicable to supervised problems, where an explicit $\mathbf{Ag}_{\{M\}}$ population is available for recognition. To overcome this limitation, a modified version of CSA for multimodal optimization problems was proposed in [7]. Its main changes are:

- We do not need to maintain a separate subset of memory $\mathbf{Ab}_{\{m\}}$, because no specific $\mathbf{Ag}_j$ has to be recognized. Instead, the whole population of antibodies will compose the memory set.
- The affinity function corresponds to the evaluation of the least-squares function, so that each Ab represents a potential solution of the problem.
- Not only the best one but also several Ab's with high affinity are selected in step 7 of the algorithm.
- There is no need maintain set $\mathbf{Ab}_{\{n\}}$, since all Ab's in the population can be selected for cloning in step 3.
- In that case, we do not need an affinity proportionate cloning. All antibodies will be cloned at the same rate (i.e., the number of clones generated for each antibody keeps the same).

5 The Proposed Method

5.1 Overview of the Method

The method presented in this paper is based on the application of the modified clonal selection algorithm to solve the optimization problem given by Eq. (9). As discussed in Section 3.2, the problem consists of solving the general problem of curve reconstruction from noisy data points with NURBS curves. To address this problem at full extent, we have to perform four different (but strongly related) tasks:

1) data parameterization,
2) breakpoint placement,
3) control point computation, and
4) weight calculation.

As a result, we have to compute the optimal values for four different sets of free variables: data parameters, breakpoints, control points, and their weights. This means that the optimization must be applied to *all* variables, not only to some of them as previous works actually did (see our discussion in Section 2). Unfortunately, the problem is more difficult than it seems at first sight because the four sets of variables are strongly intertwined in a highly nonlinear and complicated way.

Our method addresses this critical problem by applying optimization at two different stages: firstly, we consider an initial random vector of control points and apply the modified version of the clonal selection algorithm for multimodal problems to obtain optimal values for the data parameters, breakpoints, and weights. Such optimal values are subsequently used at a second stage to compute the control points. As explained before, the resulting system of equations is overdetermined, so the control points are computed through least-squares minimization by using either LU decomposition or SVD (singular value decomposition). This process is repeated until a prescribed stopping criterion is met. This final procedure yields the NURBS curve that fits the data points better in the least-squares sense.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

5.2 Description of the Method

In this section, the method is formalized and each stage of the method is discussed in detail.

5.2.1 Stage I: Optimization with the Modified CSA

In this stage, we apply the modified clonal selection algorithm to perform data parameterization, breakpoint placement, and weight computation. The input of this first stage is the set of noisy data points, $\{\mathbf{\Theta}_k\}_{k=1,\ldots,\eta}$, and the order of the approximating NURBS curve, ρ. Since all free parameters to be computed at this stage (data parameters, breakpoints, and weights) are related to the control points, we also assume an initial random vector of control points, denoted by $\{\mathbf{\Xi}_i^{(0)}\}_i$. The output of this stage is given by the data parameters, $\{\zeta_k^*\}_k$, breakpoints $\{\mu_j^*\}_j$, and weights $\{\omega_i^*\}_i$, associated with this initial population (the superscript * is used to indicate this association). This stage can be mathematically represented as:

$$\left(\{\mathbf{\Theta}_k\}_k, \{\mathbf{\Xi}_i^{(0)}\}_i, \rho\right) \longrightarrow \left(\{\zeta_k^*\}_k, \{\mu_j^*\}_j, \{\omega_i^*\}_i\right) \qquad (10)$$

However, in order to apply the modified CSA to our problem, some additional issues must be addressed, as described in next paragraphs.

Representation scheme for antibodies: Firstly, we need an adequate representation scheme for antibodies. We consider a population of individuals consisting of the vectors $\mathcal{S}_l^{(q)} = \{\mathcal{P}_l^{(q)}, \mathcal{B}_l^{(q)}, \mathcal{W}_l^{(q)}\}$, for $l = 1, \ldots, N_{ab}$, where the superscript $^{(q)}$ denotes the iteration step and N_{ab} is the total number of antibodies (i.e., the population size). The first and second components of $\mathcal{S}_l^{(q)}$, given by the real-valued vectors $\mathcal{P}_l^{(q)} = \{\zeta_{2,l}^{(q)}, \ldots, \zeta_{\eta-1,l}^{(q)}\}$ and $\mathcal{B}_l^{(q)} = \{\mu_{1,l}^{(q)}, \ldots, \mu_{\sigma,l}^{(q)}\}$, account for the data parameterization and the breakpoint vectors, respectively. They are initialized respectively with $\eta - 2$ and σ uniformly distributed random values on the interval $(0, 1)$. The elements in both vectors $\mathcal{P}_l^{(q)}$ and $\mathcal{B}_l^{(q)}$ are sorted in increasing order after initialization to meet the ordered structure of the data points and the breakpoints. Finally, the last component represents the vectors of weights $\mathcal{W}_l^{(q)} = \{\omega_{0,l}^{(q)}, \ldots, \omega_{\delta,l}^{(q)}\}$ of length $\delta + 1$, initialized with uniformly distributed random values on the interval $(0, 100)$.

Somatic hypermutation: As mentioned above, our problem is intrinsically multimodal. However, our aim is to obtain the optimal choice for the different sets of variables rather than a collection of multiple local optima. Consequently, we still keep the set $\mathbf{Ab}_{\{n\}}$ so that only the best antibodies will be cloned. The number of clones varies with the affinity, according to the following rule:

$$N_c = \sum_{j=1}^{n} round\left[(\upsilon_j + 1).\Lambda.N_{ab}\right], \qquad (11)$$

where $\upsilon_1 = 5$, $\upsilon_2 = 3$, $\upsilon_3 = 1$, and $\upsilon_j = 0$ for $j \geq 4$, and Λ is a multiplying factor. Thus, the antibodies with higher affinity will have more clones, which at turn will then undergo mutation as explained in next paragraph.

Mutation mechanism: In this work we apply a single-point, inductive uniform mutation operator for each vector $\mathcal{P}_l^{(q)}$ (respectively, $\mathcal{B}_l^{(q)}$ and $\mathcal{W}_l^{(q)}$). It introduces random perturbations on one randomly chosen component ν_i of the corresponding part of the antibody. This component is mutated according to the rule: $\nu_i \rightarrow \nu_i - \frac{\Pi}{2}\left(u - \frac{1}{2}\right)$, where $\Pi = min\left\{\frac{\nu_i}{2}, 1 - \nu_i\right\}$ and u is a random value from the distribution $U(0, 1)$. In other words, the attribute ν_i is perturbed by an additive uniform random number bounded in the neighborhood of ν_i, while all other attributes $\{\nu_j\}$ of the vector $\mathcal{P}_l^{(q)}$ (respectively, $\mathcal{B}_l^{(q)}$ and $\mathcal{W}_l^{(q)}$) remain unaltered. Finally, the elements of the vectors $\mathcal{P}_l^{(q)}$ and $\mathcal{B}_l^{(q)}$ are rearranged to keep the ordering from the previous step.

Elitism: Our algorithm is enriched by *elitism*, so that N_e of the better Ab's from the current generation are kept in the next, unaltered. This process is regulated by the parameter N_e. This feature provides faster convergence rates with respect to the non-elitist variant and improves the memory capacity of our approach.

Affinity measure: a procedure to compute the affinity is needed. In this work, the goal is to minimize the least-squares error function given by (9). However, this error function does not take into account the number of data points, so we also compute the RMSE (root mean square error) function, given by:

$$RMSE = \sqrt{\frac{\sum_{k=2}^{\eta-1}\left(\mathbf{\Theta}_k - \frac{\sum_{i=0}^{\delta}\omega_i\,\mathbf{\Xi}_i\,\varphi_{i,\rho}(\zeta_k, \mathbf{\Upsilon})}{\sum_{i=0}^{\delta}\omega_i\,\varphi_{i,\rho}(\zeta_k, \mathbf{\Upsilon})}\right)^2}{\eta - 2}} \qquad (12)$$

Finally, before searching for the solution of the problem, some control parameters should be set up. The issue of parameter tuning is explained in detail in Section 5.3. Then, the modified CSA is executed for the given number of generations. The antibody with the best (i.e., minimum) affinity value is selected as the best solution of the problem. It is represented by $\bar{\mathcal{S}}^{(q)} = \{\bar{\mathcal{P}}^{(q)}, \bar{\mathcal{B}}^{(q)}, \bar{\mathcal{W}}^{(q)}\}$.

5.2.2 Stage II: Least-Squares Data Fitting

The optimal values for the data parameters, breakpoints, and weights obtained in the previous stage are now inserted into Eq. (8) to compute the control points $\{\mathbf{\Xi}_i\}_i$. The resulting equation becomes now convex and linear, but the system is overdetermined so no explicit solution can be directly obtained. In this case, we rely on standard numerical methods, in particular, least-squares minimization. From Eq. (9), after insertion of parameters $\{\zeta_k\}_k$, $\{\mu_j\}_j$, and $\{\omega_i\}_i$, we get:

$$\underset{\{\mathbf{\Xi}_i\}_i}{min}\,||\mathbf{\Theta} - \mathbf{\Delta}.\mathbf{\Xi}||^2 \qquad (13)$$

where the optimization is now only performed on the control points $\{\mathbf{\Xi}_i\}_i$. The necessary condition for $\mathbf{\Xi}$ to be the solution of (13) is that:

$$\mathbf{\Delta}^T.\mathbf{\Delta}.\mathbf{\Xi} = \mathbf{\Delta}^T.\mathbf{\Theta} \qquad (14)$$

which leads to the *normal equation*:

$$\mathbf{\Gamma}.\mathbf{\Xi} = \mathbf{\Sigma} \qquad (15)$$

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 1
Parameters and values for the clonal selection algorithm used in our method.

Symbol	*Meaning*	*Optimal Value*	*Range of Values*
N	number of antibodies (population size)	100	$100-1000$
d	number of replaced low-affinity antibodies	20	$10-300$
n	number of antibodies to be cloned	10	$10-30$
Λ	multiplying factor in Eq. (11)	0.1	fixed
N_c	number of clones per generation	190	$190-390$
N_e	number of antibodies for elitism	10	$10-100$
N_{iter}	number of iterations for the CSA (generations)	200	$100-2000$

where $\mathbf{\Gamma} = [\mathbf{\Gamma}_{i,l}] = \left[\sum_{k=2}^{\eta-1} \xi_{l,\rho}(\zeta_k, \mathbf{\Upsilon})\xi_{i,\rho}(\zeta_k, \mathbf{\Upsilon})\right]_{i,l=0,\ldots,\delta}$ and $\mathbf{\Sigma} = [\mathbf{\Sigma}_i] = \left[\sum_{k=2}^{\eta-1} \mathbf{\Theta}_k \xi_{i,\rho}(\zeta_k, \mathbf{\Upsilon})\right]_{i=0,\ldots,\delta}$.

Note that $\mathbf{\Delta}^T.\mathbf{\Delta}$ is a symmetric square matrix and positive semidefinite, so system (15) always has a solution. Depending on its internal structure, it can be solved numerically by Gaussian elimination, LU decomposition, or by using the singular value decomposition (SVD) [38].

The algebraic solution of (14) is given by: $\mathbf{\Xi} = \mathbf{\Delta}^+.\mathbf{\Theta}$, where $\mathbf{\Delta}^+$ denotes the *Moore-Penrose pseudo-inverse* of $\mathbf{\Delta}$. Note that if the matrix $\mathbf{\Delta}$ has rank $\delta + 1$, $\mathbf{\Gamma}$ is non-singular (or equivalently, $\mathbf{\Delta}^T.\mathbf{\Delta}$ is invertible) and (15) defines $\mathbf{\Xi}$ uniquely. Furthermore, an explicit formula for $\mathbf{\Delta}^+$ is then available as $\mathbf{\Delta}^+ = (\mathbf{\Delta}^T.\mathbf{\Delta})^{-1}.\mathbf{\Delta}^T = \mathbf{\Gamma}^{-1}.\mathbf{\Delta}^T$. Note also that in this case $\mathbf{\Delta}^+$ is a left inverse of $\mathbf{\Delta}$. Otherwise, $\mathbf{\Delta}^+$ is defined by choosing $\mathbf{\Xi}$ to minimize $||\mathbf{\Xi}||^2$ among the solutions of (14).

After this procedure, the optimal vector of values for the control points, denoted as $\bar{\mathcal{C}}^{(q)}$, is obtained.

5.2.3 Iterative Process

The two previous stages can be further repeated to increase the accuracy. In particular, the following sequence is iteratively applied:

$$\begin{aligned} &\ldots \longrightarrow \left(\{\bar{\mathcal{P}}^{(q)}, \bar{\mathcal{B}}^{(q)}, \bar{\mathcal{W}}^{(q)}\}\right) \longrightarrow \bar{\mathcal{C}}^{(q)} \longrightarrow \\ &\longrightarrow \left(\{\bar{\mathcal{P}}^{(q+1)}, \bar{\mathcal{B}}^{(q+1)}, \bar{\mathcal{W}}^{(q+1)}\}\right) \longrightarrow \bar{\mathcal{C}}^{(q+1)} \ldots \end{aligned} \tag{16}$$

until a stopping criterion is met. In our case, until no further improvement of the resulting solution is achieved.

5.3 Parameter Tuning

To apply our method to the NURBS curve reconstruction problem, we need to specify two kinds of parameters, related to the approximating NURBS curve and the own method, respectively. For the former, the only input is:

1) the order of the approximating NURBS curve, ρ, and
2) the number of control points, δ,

which are freely chosen by the user. The choice of ρ is important because low-order polynomials give little flexibility in controlling the shape of the curve while high-order polynomials can introduce unwanted wiggles and require more computation. In this paper we consider third-order NURBS curves. Note, however, that our method does not depend on the value of ρ.

A more critical issue is the determination of suitable values for the parameters of the modified clonal selection algorithm. It is well known that the performance of metaheuristic methods such as the artificial immune systems is strongly dependent on the values of their parameters. Although some guidelines are given in the literature to tackle this issue, such a selection is problem-dependent and, therefore, it remains empirical to a large extent. Our approach in this paper is a combination of both strategies: good values described in the literature are used as an initial seed to guide the process; then, they are further refined through computer experiments. To this purpose, we have analyzed experimentally the effects of the variation of parameter values on the performance of the method and on the quality of the solution. As a result, we have derived the set of parameter values reported in Table 1. The table shows four different items (in columns): the symbol of the parameter, its meaning, its selected value for this problem, and the range of parameter values we tested. The selected values are optimized as they provide a very accurate solution to our problem while minimizing the computational cost. The parameters analyzed in our study are:

- population size, N_{ab}: it is well known that a certain population size is required to promote exploration of the search space in evolutionary algorithms. In the case of AIS, some previous works suggested a minimum of 100 antibodies for better efficiency. On the other hand, larger values of N_{ab} increase the exploration capacity but also the number of function evaluations (and hence, the runtime). In this paper, we tested our method for population sizes ranging from 100 to 1000 antibodies with step-size 100, and obtained very similar results in all cases. Since larger populations imply longer computation times without any error improvement, a value of $N_{ab} = 100$ is recommended in this work.
- parameter d: this number describes the amount of low-affinity antibodies replaced during the execution to increase the exploration ability of the method. Its choice is troublesome, since too low (high) values degrade the exploration (exploitation) capabilities of the method, respectively. In this paper, we tested different values for this parameter ranging from 10 to 300 with step-size 10. Our computer simulations show that the fitting error improves from 10 to 20,

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

but no significant improvement is reached for higher values. Therefore, we set this value to $d = 20$.

- parameter n: it represents the number of antibodies to be cloned. This number is usually a percentage of the population size, typically ranging from 10% to 30%, so we tested values for this parameter from 10 to 30 with step-size 10. We obtained similar fitting errors in our simulations for all cases but larger values of n mean longer runtimes and more storage and capacity requirements, so we set this value to $n = 10$.
- Λ: it is a multiplying factor in Eq. (11) combined with the population size. Therefore, its value can be fixed during the computation process of the algorithm. In this paper, its value is set to $\Lambda = 0.1$.
- parameter N_c: it represents the number of clones generated at each new generation. According to Eq. (11), it is determined by the values of previous parameters N_{ab}, n, and Λ. By doing so, the value of this parameter ranges from 190 to 390 clones. The results of our method do not change when we increase the value of N_c over this entire range, but complexity increases, so we take the lowest value $N_c = 190$.
- parameter N_e: it represents the number of antibodies selected for elitism (i.e., the memory of the process). In this paper, we tested different values for this parameter from 10 to 100 with step-size 10, but we did not notice any significant difference in our results, so we decided to keep this parameter to the minimum value $N_e = 10$ in order to save computational time.
- number of iterations,N_{iter}: this is a critical parameter, because we need to ensure we reach convergence without wasting unnecessary resources (i.e., iterations) in the process. Once again, the optimal value was determined empirically; the parameter was varied in the range 100 to 2000 iterations with step-size 100. In our trials, 200 iterations are enough to reach convergence. This is the value used in this paper.

After this parameter tuning, our algorithm is executed for the chosen number of iterations. The antibody with the best (i.e., minimum) affinity value is selected as the best solution to the problem.

5.4 Computational Complexity

Regarding the complexity of the method, the main computational load is due to the affinity calculation, resulting in $O(N_{ab}.(\eta + \delta + \sigma - 1))$ overall, selection and re-selection of the highest-affinity antibodies (requiring $O(N_{ab})$ and $O(N_c)$ in the worst cases, respectively) and hypermutation, requiring $O(N_c.(\eta + \delta + \sigma - 1))$ in our case. Therefore, the total computational time is of order $O((N + N_c).(\eta + \delta + \sigma - 1))$. If the process is repeated iteratively during q cycles, the total cost is $O(q.(N+N_c).(\eta+\delta+\sigma-1))$. Finally, the computation of the control points for the data fitting stage using either SVD or LU decomposition requires $O(\delta^3)$.

6 Experimental Results

Our method has been applied to the reconstruction of four outline curves in medical imaging from sets of noisy data points. The first example, displayed in Figure 2, corresponds to a color image of a MRI brain, shown on the left. From this image, we obtained a list of 1558 scanned data points. The scanning procedure is mostly optical, based on the color map of the figure, and hence, some noise is introduced during the process. The application of our method to this data set yields the third-order NURBS curve that approximates the data points better in the least squares sense. It is displayed in Fig. 2 (middle). As the reader can see, the reconstructed curve (labelled as *Curve I* onwards) exhibits a very complicated shape, with several concavity changes and turning points. Even in this case, our method is able to recover the underlying shape of the data points with very good quality. This good visual quality is confirmed by our numerical results: we have obtained a RMSE of 8.778146×10^{-4}. Figure 2 (right) shows the curve superimposed on the original image for better visualization. This example shows the ability of our method to replicate intricate shapes with high accuracy.

The second example, displayed in Figure 3, corresponds to a gray tone image of a MRI brain, shown on the left. In this case, three different closed outline curves - corresponding to boundaries between different tissues going from the innermost to the outermost curve - are to be reconstructed. They are labelled as *Curve II*, *Curve III*, and *Curve IV* and displayed in Figure 3 (middle) in different colors: blue, green, and red, respectively. They have been obtained through automatic optical scanning with 778, 258, and 268 data points, respectively. The RMSE of the three curves are: 6.687015×10^{-4}, 4.283281×10^{-4}, and 6.768319×10^{-4}, respectively. Note that the geometry of the curves is more irregular than it appears a first sight. This fact, common for the three curves, is more noticeable visually for the blue and red ones, as they contain many small regions with turning points and sharp changes of direction and concavity. Still, the method obtains very good results in all cases.

6.1 Implementation Issues

All computations in this paper have been performed on a 2.6 GHz. Intel Core i7 processor with 8 GB. of RAM. The source code has been implemented by the authors in the native programming language of the popular scientific program *Matlab*, version 2013b. In our opinion, *Matlab* is a very suitable tool for this task: it is fast and provides reliable, well-tested routines for efficient matrix manipulations. It also contains a bulk of resources regarding the solving of systems of equations. This feature proved to be very valuable in case of ill-conditioned matrices, i.e., with too large (or even infinite) condition number. This is a situation that can actually happen in practice, for instance, when one or several singular values in SVD decomposition are null or very near to zero. Advisable answer to this problem is to set reciprocals of such singular values to zero. *Matlab* command `svd` handles this situation for us. Similarly, when LU decomposition is used instead, *Matlab* command `lu` returns a suitable matrix factorization regardless the sparsity of the matrix. Also, *Matlab* provides the command `mldivide` to solve the equation (14) through specialized *Matlab* commands for Gaussian elimination with partial pivoting and singular value decomposition (SVD) for squared and non-squared systems, respectively. Depending on the general

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

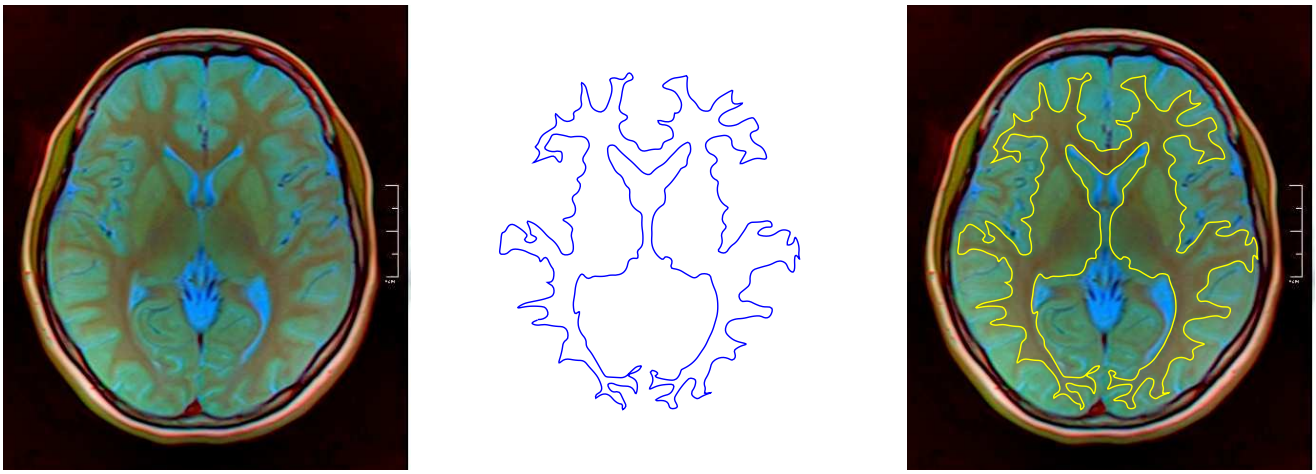

Fig. 2. Application of our method to a brain MRI: (left) original MRI; (middle) reconstructed NURBS curve; (right) combination of both images for better visualization. (Figure on the left: ©Nevit Dilmen. Used under Creative Commons license, BY-SA 3.0).

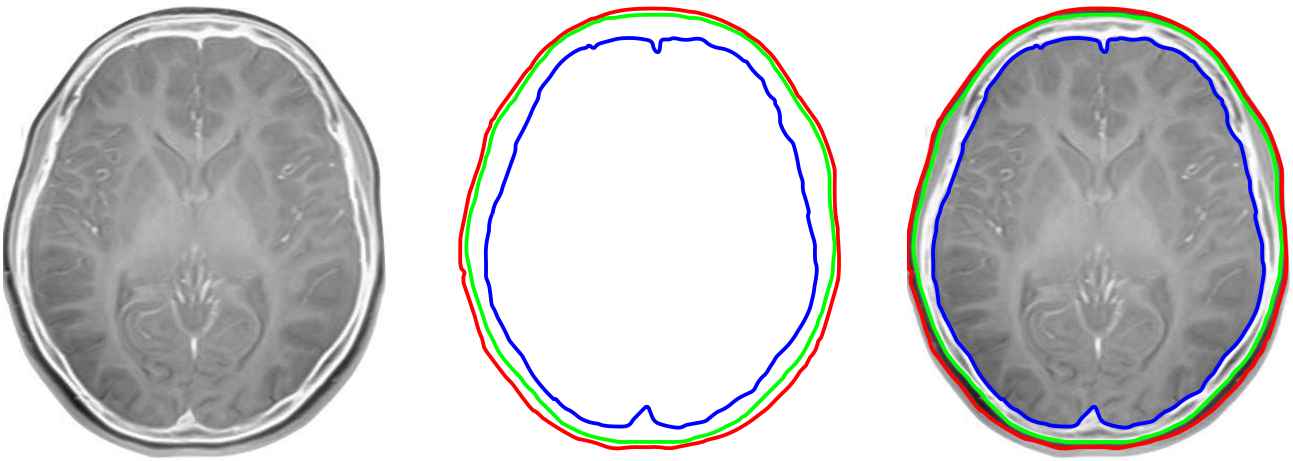

Fig. 3. Application of our method to a brain MRI: (left) original MRI; (middle) three reconstructed NURBS curves corresponding to the boundaries of different tissues; (right) combination of both images for better visualization. (Figure on the left: ©Nevit Dilmen. Used under Creative Commons license, BY-SA 3.0).

structure of matrix $\mathbf{\Gamma}$, this command applies specialized LAPACK and BLAS routines to get the best possible solution to this system. The interested reader is referred to [38] for a detailed description of all these numerical routines and their efficient computer implementation.

6.2 Computational Execution Time

Regarding the computational times, it is well-known that metaheuristic techniques are relatively slow and time-consuming. The CPU times depend on many factors, such as the complexity of the model, number of free parameters, population size, number of iterations, computer and programming language used for execution, and others. This makes difficult to determine in advance the execution runtime for a given example. For illustration, with the configuration described in this paper, it took about 4-6 minutes for the first example, and between 2-4 minutes for the other three examples. Although the CPU times that we obtained make the method not well suited for real-time applications, they are still competitive and affordable for many applications. They are not surprising either, considering the large number of free variables of the problem. We remark, however, that the emphasis of this method is not on the computational speed, but on the accuracy of results.

For further assessment, we also implemented our method in C++ and Java and compared their CPU times with those of our code in *Matlab*. In our results, we found that the execution in Java is much slower than in *Matlab*; on the contrary, the code in C++ is slightly faster, but not significantly. This can be explained by the fact that our problem is very well suited for matrix representation and handling, and matrices are exactly the fundamental representation of information and data in *Matlab*. In general, the routines in *Matlab* are extremely optimized for matrix handling, something that does not happen in C++.

7 Comparison With Other Approaches

As discussed in Section 2, despite the great interest and applicability of the curve reconstruction problem with B-splines, most of the previous work reported in the literature

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 2
RMSE fitting errors with different methods for the four examples in this paper (best results are highlighted in bold).

Method	Curve I	Curve II	Curve III	Curve IV
uniform/uniform	7.016562E+1	2.601976E+1	3.514852E0	4.799686E0
uniform/arc-length	6.779406E0	1.763314E0	3.326017E-1	5.346517E-1
uniform/centripetal	4.831002E0	9.558726E-1	1.091571E-1	3.824439E-1
averaging/uniform	3.363111E0	9.875109E-1	4.692388E-2	8.836551E-1
averaging/arc-length	9.834209E-1	7.782215E-1	3.343125E-2	1.131625E-1
averaging/centripetal	8.177386E-1	4.668482E-1	2.917348E-2	9.586774E-2
ClonalG/centripetal [47]	1.235649E0	8.002491E-1	5.552783E-1	7.631058E-1
PESA [48]	3.953788E-1	9.510394E-2	3.918248E-2	5.114736E-2
averaging/PSO [44]	7.622518E-2	4.072311E-2	3.668523E-3	4.110793E-3
this method	**8.778146E-4**	**6.687015E-4**	**4.283281E-4**	**6.768319E-4**

is focused on simple cases, such as the explicit curves. For the parametric case, only partial solutions are described, based on computing only some of the free variables of the problem, either the data parameters or the breakpoints. To the best of our knowledge, no technique has addressed the general B-spline curve reconstruction problem so far, much less the more difficult case of NURBS curves. This fact is not accidental, but a clear indication of the difficulty of this task.

This lack of previous references prevents any detailed comparison with other general methods. Yet, it is possible to make a comparison with other methods commonly accepted in the field, even although they provide only partial solutions. This includes the three classical parameterization methods indicated in Section 2: uniform, arc-length, and centripetal, along with two methods for breakpoint placement: uniform and averaging. Since solving the general case requires the computation of both data parameters and breakpoints, these partial approaches have to be combined, yielding six feasible couples, given by the methods for breakpoint placement and data parameterization, respectively. They are listed in rows 1–6 in Table 2.

Other three methods for parametric B-spline curves (listed in rows 7–9 of Table 2) can be found in the literature. It is important to remark, however, that these methods do not include weights; therefore, they cannot compute NURBS but only strictly polynomial B-spline curves. The first one applies an artificial immune system approach to compute a breakpoint allocation from a centripetal parameterization [47]. This immune approach is replaced in the second method by a Pareto Envelope-Based Selection Algorithm (PESA), a type of genetic algorithm well suited for multi-objective optimization [48]. The third method computes the breakpoints by the averaging method, while particle swarm optimization (PSO) is applied for the data parameters [44]. Finally, the method proposed in this paper is considered in row 10.

Table 2 reports the RMSE values for all these methods (in rows) and the four examples (in columns) of this paper. Best results are highlighted in bold for easier identification. As the reader can see, the proposed method outperforms all other methods for all instances in the benchmark. Among the classical methods, the uniform parameterization is generally the worst. This is not surprising, since the data points are not uniformly distributed. We can also see that the centripetal parameterization is slightly better than the arc-length, but they still perform very similarly. The slight advantage of the centripetal parameterization is more noticeable for Curves I and II, which exhibit sharp turns. On the other hand, the averaging method for breakpoint placement performs better than the uniform method for all instances in the benchmark. This effect is due to the several changes of concavity, which required more breakpoints to be located around these areas for better fitting.

Although based on a powerful metaheuristic technique, the method in [47] does not improve the classical methods. This can be explained by the fact that the method is not continuous, but based on the conversion of the original continuous optimization problem into a combinatorial optimization problem. This conversion process introduces large discretization errors, even for very simple shapes. This discrete approach is also applied in [48], although the fitting errors improve in all cases. The method in [44] performs reasonable well: it improves the results of the classical methods and those in [47], [48] significantly. Still, its results are much worse than those with our method, about one or two orders of magnitude, depending on the given example. This result can be explained by the fact that the method in [44] is only suitable for B-splines, but not for NURBS. Consequently, it lacks the flexibility given by the extra degrees of freedom provided by the weights. As a conclusion, the method introduced here outperforms previous methods in the literature by orders of magnitude on the examples of our benchmark.

8 Conclusions and Future Work

This paper addresses the problem of NURBS curve reconstruction from sets of noisy data points with application to the case of outline curves of medical images. The relevance of this work comes from the fact that this is *the first method solving the full problem of curve reconstruction with NURBS*. By "full" we mean to compute all relevant free variables of the problem. In that case, we are dealing with a difficult non-convex, nonlinear, high-dimensional, multimodal, continuous optimization problem. Our method is based on a combination of two techniques: an immunological approach – called modified clonal selection algorithm – to perform data parameterization, breakpoint placement, and weight calculation, and least squares minimization to compute the control points. This procedure is repeated iteratively (until no further improvement is achieved) for higher accuracy. Our analysis also include important issues such as a proper parameter tuning for better performance, the computational complexity, some implementation issues, and the computational time. The method has been applied to reconstruct

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

four outline curves of different complexity from MRI brain images with excellent results. Although there is no previous method solving the full problem with NURBS, we have still carried out a comparative analysis with other related approaches in the field for completeness. Our computational experiments show that our method outperforms all previous approaches reported in the literature by orders of magnitude for the examples described in this paper.

Main limitations of this method are the computation time and the determination of the optimal order for the NURBS curve. The first problem might arguably be partially alleviated by parallelization, but it is still an open question to determine the rate of improvement achieved with this modification (if any). The determination of the optimal value for the order would probably require to modify the fitness function by introducing some kind of penalty term modulating this factor. We tried some very preliminary ideas on this problem with little success so far. On the other hand, one of the reviewers suggested us to apply some kind of self-adaptation scheme to our algorithm in order to improve its performance. We find this idea promising and would like to explore this approach in detail. All these issues will be part of our future work in the field.

Acknowledgments

This research has been kindly supported by the Computer Science National Program of the Spanish Ministry of Economy and Competitiveness, Project Ref. #TIN2012-30768, and the Faculty of Sciences of Toho University, Funabashi, Japan. Special thanks are owed to the editor and the three anonymous reviewers for their encouraging and constructive comments and suggestions and very helpful feedback that allowed us to improve our paper.

References

[1] 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) (2001), 1–16.

[2] Barnhill, R.E.: *Geometric Processing for Design and Manufacturing*. SIAM, Philadelphia (1992)

[3] de Boor, C. A.: *Practical Guide to Splines*. Springer-Verlag (2001)

[4] De Castro, L.N., Timmis, J.: *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, London (2002)

[5] De Castro, L.N., Von Zuben, F.J.: *Artificial Immune Systems: Part I - Basic Theory and Applications*. Technical Report-RT DCA 01/99 (1999)

[6] De Castro, L.N., Von Zuben, F.J.: The Clonal Selection Algorithm with Engineering Applications. In: *Proceedings of GECCO00, Workshop on Artificial Immune Systems and their Applications*, Las Vegas, USA, pp. 36–37 (2000)

[7] De Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Computation*, 6(3), pp. 239–251 (2002)

[8] Dasgupta, D. (editor): *Artificial Immune Systems and Their Applications*. Springer Verlag, Berlin (1999)

[9] Dasgupta, D., Nino, F.: *Immunological Computation: Theory and Applications*. CRC Press, Boca Raton, FL (2008).

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

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

[12] Fister Jr., I., Yang, X.S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. *Elektrotehniski Vestnik*, 80(3), pp. 116–122 (2013).

[13] Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. *Proceedings of IEEE Symposium on Research in Security and Privacy*. Los Alamitos, CA, pp. 202–212 (1994).

[14] Gálvez, A., Iglesias, A.: Efficient particle swarm optimization approach for data fitting with free knot B-splines. *Computer Aided Design*, 43(12), pp. 1683–1692 (2011).

[15] Gálvez, A., Iglesias, A.: Iterative two-step genetic-algorithm method for efficient polynomial B-spline surface reconstruction. *Information Sciences*, 182(1), pp. 56–76 (2012).

[16] Gálvez, A., Iglesias, A.: Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Information Sciences*, 192(1), pp. 174–192 (2012).

[17] Gálvez A., Iglesias A.: A new iterative mutually-coupled hybrid GA-PSO approach for curve fitting in manufacturing. *Applied Soft Computing*, 13(3) (2013), 1491–1504.

[18] Gálvez A., Iglesias A.: Firefly algorithm for explicit B-Spline curve fitting to data points. *Mathematical Problems in Engineering*, Article ID 528215 (2013), 12 pages.

[19] 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*, Article ID 283919 (2013), 10 pages.

[20] Gálvez, A., Iglesias, A.: Particle-based meta-model for continuous breakpoint optimization in smooth local-support curve fitting. *Applied Mathematics and Computation*, 275, pp. 195–212 (2016)

[21] Gálvez A., Iglesias A.: New memetic self-adaptive firefly algorithm for continuous optimization. *International Journal of Bio-Inspired Computation*, 8(5) (2016) 300–317.

[22] Gálvez A., Iglesias A., Avila, A.: Hybridizing mesh adaptive search algorithm and artificial immune systems for discrete rational Bézier curve approximation. *The Visual Computer*, 32, pp. 393–402 (2016)

[23] 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, pp. 90–106 (2015)

[24] Goldenthal, R., Bercovier, M.: Spline curve approximation and design by optimal control over the knots. *Computing*, 72 (2004) 53-64.

[25] Greensmith, J., Aickelin, U.: Artificial dendritic cells: multi-faceted perspectives. *Proc. of Human-Centric Information Processing Through Granular Modelling*, (2009) 375–395.

[26] Gu, P., Yan, X.: Neural network approach to the reconstruction of free-form surfaces for reverse engineering. *Computer-Aided Design*, 27(1) (1995), 59–64.

[27] Hart, E.,Timmis, J.: Application areas of AIS: past, present and future. *Applied Soft Computing*, 8(1) (2008) 191–201.

[28] Hoffmann M.: Numerical control of Kohonen neural network for scattered data approximation. *Numerical Algorithms*, 39 (2005), 175–186.

[29] Iglesias, A., Echevarría, G., Gálvez, A.: Functional networks for B-spline surface reconstruction. *Future Generation Computer Systems*, 20(8) (2004), 1337–1353.

[30] Iglesias, A., Gálvez, A.: Hybrid functional-neural approach for surface reconstruction. *Mathematical Problems in Engineering*, Article ID 351648 (2014), 13 pages.

[31] Jacobson, T.J., Murphy, M.J.: Optimized knot placement for B-splines in deformable image registration. *Medical Physics*, 38(8) (2011) 4579–4592.

[32] Jing, L., Sun, L.: Fitting B-spline curves by least squares support vector machines. *Proc. of the 2nd. Int. Conf. on Neural Networks & Brain*. Beijing (China). IEEE Press, 2005, pp. 905–909.

[33] Jupp, D.L.B.: Approximation to data by splines with free knots. *SIAM Journal of Numerical Analysis*, 15 (1978) 328–343.

[34] Li, W., Xu, S., Zhao, G., Goh, L.P.: Adaptive knot placement in B-spline curve approximation. *Computer-Aided Design*, 37 (2005), 791–797.

[35] Ma, W.Y., Kruth, J.P.: Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer Aided Design*, 27(9) (1995), 663–675.

[36] The Neiman Report: *Medical Imaging: Is the Growth Boom Over?* Brief #1 (2012).

[37] Patrikalakis, N.M., Maekawa, T.: *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer Verlag (2002)

[38] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes (2nd. edition)*, Cambridge University Press, Cambridge, 1992.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2688444, IEEE/ACM Transactions on Computational Biology and Bioinformatics

[39] Jerne, N. K.: Toward a network theory of the immune system. *Ann. Immunol.*, 125 (1974) 373–389.
[40] Park, H.: An error-bounded approximate method for representing planar curves in B-splines. *Computer Aided Geometric Design*, 21 (2004), 479–497.
[41] Park, H., Lee, J.H.: B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design*, 39 (2007), 439–451.
[42] Piegl, L., Tiller, W.: *The NURBS Book*. Springer Verlag, Berlin/Heidelberg (1997)
[43] Sarfraz, M., Raza, S.A.: Capturing outline of fonts using genetic algorithms and splines. *Proc. of Fifth International Conference on Information Visualization IV'2001*, IEEE Computer Society Press, 738–743 (2001)
[44] Sun, Y.H., Tao, Z.L., Wei, W.X., Xia, D.S.: B-spline curve fitting based on adaptive particle swarm optimization algorithm. *Applied Mechanics and Materials*, 20-23 (2010) 1299-1304.
[45] The Guardian: *World's largest medical imaging study will scan 100,000 Britons*. Web site: *https://www.theguardian.com/science/2016/apr/14/worlds-largest-medical-imaging-study-will-scan-100000-britons*. (Last retrieved on Sept. 19 2016).
[46] Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. *Theoretical Computer Science*, 403(1) (2008) 11–32
[47] Ulker, E., Arslan, A.: Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Information Sciences*, 179 (2009) 1483-1494
[48] Ulker, E.: B-Spline curve approximation using Pareto envelope-based selection algorithm-PESA. *International Journal of Computer and Communication Engineering*, 2(1) (2013), 60–63.
[49] Varady, T., Martin, R.R., Cox, J.: Reverse engineering of geometric models - an introduction. *Computer Aided Design*, 29(4), (1997) 255–268.
[50] 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) (2006), 214–238.
[51] Yang, X.S.: *Nature-Inspired Metaheuristic Algorithms (2nd. Ed.)*. Luniver Press, Frome, UK (2010).
[52] Yang, X.S.: *Nature-Inspired Computation in Engineering*. Studies in Computational Intelligence, vol. 637, Springer International Publishing Switzerland (2016).
[53] 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, 1999, pp. 162–169.
[54] Yoshimoto F., Harada T., Yoshimoto Y.: Data fitting with a spline using a real-coded algorithm. *Computer Aided Design*, 35 (2003) 751–760.
[55] Zhao, X., Zhang, C., Yang, B., Li, P.: Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation. *Computer Aided Design*, 43 (2011) 598–604.

Andrés Iglesias belongs to the Department of Information Science, Toho University, Funabashi, Japan, and the Department of Applied Mathematics and Computational Sciences, University of Cantabria, Santander, Spain. He has also had research stays at Wessex Institute of Technology, Southampton, UK, International Center for Theoretical Physics, Trieste, Italy, and University of Tsukuba, Tsukuba, Japan. He has published more than 200 international journal and conference papers and 14 books. He has been chairman and organizer of more than 50 international conferences, program committee member of more than 200 international conferences, and editorial board member of several international journals. He has been a reviewer of more than 130 journal papers (mostly in JCR) and more than 400 conference papers, and scientific expert evaluator of projects for NSF (USA), the European Union, and several national and regional public research agencies. His research interests include computer graphics, geometric modeling, artificial intelligence, soft computing, and their applications.

Akemi Gálvez belongs to the Department of Information Science, Toho University, Funabashi, Japan, and the Department of Applied Mathematics and Computational Sciences, University of Cantabria, Santander, Spain. She has published more than 100 international papers in scientific journals and conferences, including papers in Q1 journals in eight categories of JCR-Science citation index. She has been a reviewer and program committee member of several international conferences and editorial board member of five international journals. She has participated in several national and international research projects. Her research interests include geometric modeling and processing, artificial intelligence, soft computing, scientific programming and visualization, and their applications.

Andreina Avila is currently a PhD candidate at the Department of Applied Mathematics and Computational Sciences, University of Cantabria, Santander, Spain. She holds a BSc degree in Computer Science from the University of Zulia, Maracaibo, Venezuela, in 2008 and a MSc degree on Computer Science from the University of Cantabria in 2012. She has published some papers on artificial intelligence and soft computing for curve and surface reconstruction in international journals and conferences. Her main research fields are geometric modeling, artificial intelligence, curve and surface reconstruction, and their industrial and medical applications.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.