An overview of constrained fitting optimization techniques for reverse engineering of mechanical parts

Francesco Buonamici, and Alessandro Lapini

***Abstract*—** Reverse Engineering represents a long-term goal of computer science and engineering; it aims at the reconstruction of digital models from measured data by means of 3D mathematical surfaces and geometrical features representing the geometry of a physical part. In this paper, an overview of constrained fitting optimization methods specifically devised for the reconstruction of mechanical parts is proposed, highlighting the connections between the theoretical problem and some practical solutions. Furthermore, algorithmic procedures are provided in order to underline the main differences between the considered approaches. Critical aspects of constrained fitting and recent trends on Reverse Engineering are finally presented and discussed.

***Keywords*—** reverse engineering; constrained fitting; CAD reconstruction; optimization algorithms.

I. Introduction

The reconstruction of digital geometric models of physical objects has been one of the most studied topics of the Computer Aided Design (CAD) field for the past 20 years. The interest in Reverse Engineering (RE) solutions has been recently strengthened as the area of potential applications has widened, going beyond the traditional mechanical engineering field (e.g. medical [1], topology optimization [2], archeology [3], cultural heritage [4], [5]).

The vast majority of CAD reconstruction processes rely on the framework depicted in Figure 1 [6], the first step being the acquisition of 3D data (points, mesh) on the surfaces of the object to be reconstructed; occasionally, a different information source can be used to extract the reference data guiding the reconstruction of the 3D model [7]. The acquired data is subsequently processed, to identify and extract the surfaces composing an object; this phase is usually carried out analyzing local geometric properties of the acquired data, e.g. differential geometry. At the end of the processing, an analytical description, i.e. a digital geometric model, is produced.

In mechanical applications, specifically, the whole process is oriented towards the generation of a proper parametric CAD model. The composing CAD features are required to be correct in dimensions, topology and in the existing relations between them.

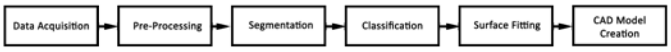

Figure 1 – Traditional RE framework.

The reconstruction result needs to be correct with respect to both the retrieved topology (i.e. which CAD features compose the object, the geometric relations and regularities enforced) and the dimensional accuracy. The retrieval of the original design intent of a mechanical part and the practical usefulness of the reconstructed model highly rely on the imposition of a valid set of geometric constraints. The correct functioning of mechanical parts, in fact, often depends on geometric relations between functional surfaces or features (e.g. parallelism of two planes, orthogonality between axes, etc.) and their retrieval is in most cases fundamental [8].

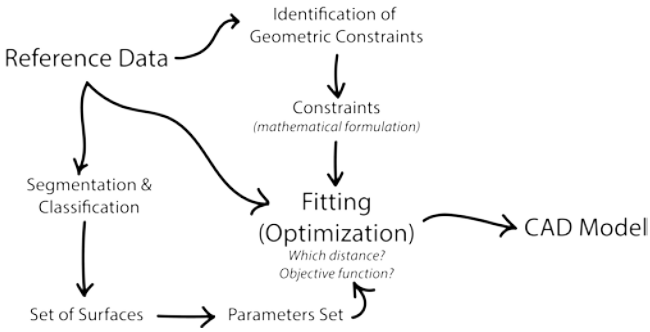

Figure 2 – Constrained Fitting Framework

The most pursued strategy to achieve the described goals is the so called *constrained fitting* approach [9]. This class of methods introduce a set of geometrical constraints that are defined among the identified geometric features and enforced by formulating a constrained optimization problem. Basic steps composing the typical constrained fitting approach are presented in Figure 2: summarizing the whole process, three elements are taken as input in the optimization steps: reference data, geometric constraints and a set of parameters (i.e. the optimization variables) describing the shape of the surfaces

F. Buonamici is with the Department of Industrial Engineering of Florence, University of Florence, Via di Santa Marta 3, 50139 Florence, Italy.

A. Lapini is with the Department of Industrial Engineering of Florence, University of Florence, Via di Santa Marta 3, 50139 Florence, Italy. (corresponding author to provide phone: +39 055 275 8687; fax: +39 055 275 8755; e-mail: alessandro.lapini@unifi.it).

constituting the final model. The definition of a coherent and solvable constraint set, and the implementation of an effective and efficient optimization routine represent one of the most difficult and studied aspects of the constrained fitting approach.

Several contributions have been discussed in literature to tackle these problems, but an up-to-date survey of the optimization techniques implemented to solve these problems is still a missing point. Accordingly, the aim of the present paper is to address this issue, presenting an overview of the optimization algorithms and approaches that have been proposed to solve constrained fitting problems in the RE field. In Section II, the formalization of the constrained fitting problem is presented, providing insights into practical solutions proposed in the literature. In Section III and Section IV, gradient based and evolutionary optimization algorithms are, respectively, introduced, focusing on their application for constrained fitting problems. Final considerations are drawn in Section V.

II. CONSTRAINED FITTING PROBLEM IN REVERSE ENGINEERING

A. Formulation

As a general starting point of the considered framework, we assume that a pre-segmented mesh extracted from acquired 3D data of an object is available [8], [10], [11]. Specifically, each point p_{ij} is associated to a surface s_i that has been previously detected by some specific segmentation algorithm [12], [13] ; the index j accounts for all the points linked to that surface.

Since we are dealing with parametric surfaces, the i-th surface is completely defined by a column vector of parameters $\mathbf{x_i}$, whose elements depend on the kind of the considered surface and on the adopted mathematical description. Hence, if N is the total number of surfaces, the parameter vector describing all the object's surfaces is $\mathbf{x} = [\mathbf{x_1}^T, \mathbf{x_2}^T, \dots \mathbf{x_N}^T]^T$, being $()^T$ the transpose operator.

Moreover, let's assume that a set of constraints involving the elements of $\mathbf{x}$ are specified or automatically extracted from the acquired data by a suitable constraints-detecting procedure. Constraints play a fundamental role in the framework; more insights will be given below.

The aim of the constrained fitting approach in RE is to find the *best* set of parameters $\mathbf{x}$, such that i) the reconstructed object's surfaces are as *close* as possible to the acquired data, ii) the constraints on parameters are satisfied. The problem of the constrained fitting can be formulated as

$$\begin{aligned} &\textbf{minimize} \quad F(\mathbf{x}) \\ &\textbf{subject to contraints} \quad g_k(\mathbf{x}) \le 0, \quad k = 1, \dots M_i \\ &\qquad\qquad\qquad\quad\; c_k(\mathbf{x}) = 0, \quad k = 1, \dots M_e. \end{aligned} \tag{1}$$

In (1), F(x) is namely the objective function that accounts for the unconstrained fitting error, that is, the discrepancies between the parametric surfaces and the acquired data; g_k and c_k are the k-th inequality and equality constraint, respectively, that involve one or more parameters, potentially belonging to more surfaces.

According to the previous formulation, the constrained fitting for RE is a global optimization problem (as highlighted by the attempts at solving the problem with a local optimization algorithm discussed in [14]) that requires a balancing between the approximation and the constraint satisfaction. In the following subsections, a discussion on suitable choices for the objective function and the constraints is provided, considering both the RE and the mathematical point of view.

B. Objective function

The rationale behind the objective function is to *measure* how far the parametric surfaces are from the points of the dataset, according to some measure function. Consequently, the aim is to reduce such distance in order to improve the fidelity of the reconstructed object to the data.

Algebraic metric has been initially adopted in the literature for the fitting of quadric surfaces [11]. The concept can be introduced by generally considering a polynomial description of the i-th surface:

$$\mathbf{h_{ij}^T x_i} = 0\,, \tag{2}$$

where $\mathbf{h_{ij}}$ is the measurement vector whose element are given by monomial of coordinate of the acquired point p_{ij}. For instance, if $p_{ij} = (x,y,z)$ is a point associated to plane then $\mathbf{h_i}=[x,y,z,1]^T$ and $\mathbf{x_i}=[u,v,w,d]$. In practice acquired data are affected by noise; thus, the right member of (2) can be generally substituted by an offset term $\varepsilon_{ij} \neq 0$. If m_i measurements are provided, then local algebraic fitting of the i-th surface is formulated as a (ordinary) least squares problem over all the ε_{ij}; the functional to be minimized is

$$f_i(\mathbf{x_i}) = \sum_{j=1}^{m_i} \varepsilon_{ij}^2 = \mathbf{x_i}^T \mathbf{H_i x_i}\,, \tag{3}$$

where the element of the square matrix $\mathbf{H_i}$ in position (l,q) is

$$\mathbf{H_i}(l,q) = \mathbf{h_{il}^T h_{iq}}\,. \tag{4}$$

By extension, the objective function related to algebraic fitting of the whole object is

$$F(\mathbf{x}) = \sum_{i=1}^{N} \omega_i f_i(\mathbf{x_i}) = \mathbf{x}^T \mathbf{H x}, \tag{5}$$

being $\mathbf{H}$ a diagonal block matrix containing the local fitting matrices $\omega_i\mathbf{H_i}$. Please note that the term ω_i is an eventual positive weighting factor to enforce priority fitting on i-th surface.

The main advantage of using (5) is given by its quadratic form, which is in practice strictly convex; by imposing simple normalization on $\mathbf{x_i}$ to avoid trivial solutions (e.g., $a^2 + b^2 +$

c^2= 1 in the case of a plane), this leads to an eigenvalue problem whose solution can be managed by efficient and numerically stable algorithms. Furthermore, the system matrix **H** is sparse and can be computed only once, preferably in an off-line stage. Werghi [9], [15] has also given a statistical interpretation of algebraic fitting as the likelihood function of range data over the parameters if the offset term ε_{ij} is modelled as additive Gaussian and its statistics are known, a weighted least squares approach can be followed by inserting appropriate scalar weights in (4).

Even though algebraic fitting is appealing for its ease to be solved, it is not linearly related to *geometrical* distance between surfaces and points. Indeed, the term ε_{ij} only accounts for the overall algebraic offset from 0 due to the noise affecting the point's coordinates; such an offset is a function of the measurement vector, that is, of polynomial combinations of point's coordinates. Since ordinary least squares method does not consider how the noise affects each coordinate, the solution qualitatively tends to compensate more for coordinates with higher polynomial degrees, even when the acquisition error is homogeneous for all coordinates. This leads to significant shape bias when considering simply curved surfaces [11], which cannot be acceptable for RE purposes.

A straightforward alternative to algebraic distance is the *Euclidean* distance:

$$f_i(\mathbf{x_i}) = \sum_{j=1}^{m_i} \left\| p_{ij} - s_i(\mathbf{x_i}) \right\|^2 , \tag{6}$$

where $s_i(x_i)$ denotes that the surface depends on the associated parameters vector. It should be noted that Euclidean metric and algebraic one generally differ, being linear structures (lines and planes) noticeable exceptions. Form an operative point of view, the Euclidean distance requires to find the closest point on a surface for each acquired point; hence, it deals with geometric distances and represents a suitable solution for RE application. Nevertheless, (6) is generally a non-convex function; hence, no general closed form solution for the minimization of (6) is available, even for quadric surfaces [11]. Optimization problems dealing with non-convex functionals are encountered across several engineering fields [16], and sub-optimal solving algorithms are usually and possibly developed to avoid unfeasible computational costs. As a simple example, let's consider the case of the Euclidean distance between a sphere having parameters $\mathbf{x_i} = [r,x_0,y_0,z_0]^T$ (r is the sphere radius, being the remaining parameters the center's coordinate) and generic points $p_{ij} = (x_j,y_j,z_j)$; it yields

$$f_i(r, x_0, y_0, z_0) = \sum_j \left(\sqrt{(x_j - x_o)^2 + (y_j - y_o)^2 + (z_j - z_o)^2} - r \right)^2 . \tag{7}$$

The objective function in (7) is non-convex; hence, its constrained minimization is a non-trivial task.

Euclidean distance has been less considered in the RE field due to its mathematical complexity. Some efficient implementations have been proposed [17], [18]; alternative measures have been also investigated in the literature for fitting purposes, such as Taubin distance [19], or for similarity detection, such as Haussdorf distance [20].

In order to improve the tractability of the mathematical problem as well as the computational efficiency of solving algorithms, the concepts of *faithful* distances and *efficient* representation have been introduced and widely exploited in the RE community [8], [10], [21].

The rationale behind the faithful distance follows from the fact that the exact measure of the Euclidean distance between a surface and a point for every reciprocal position is not strictly necessary, but increasing accuracy is required as they get closer. Hence, a distance is faithful to the Euclidean one if [8] i) is zero when the point is on the surface, ii) its derivative on the surface is unit and orthogonal to the surface. Since faithful distance relaxes the original metric, there exist several possibilities for a given scenario and the most suitable ones can be accordingly chosen. As to previous example, a faithful distance for (7) is

$$f_i(r, x_0, y_0, z_0) = \sum_j \frac{(x_j - x_o)^2 + (y_j - y_o)^2 + (z_j - z_o)^2 - r^2}{2r} . \tag{8}$$

It is worthy to note that algebraic distances are commonly also faithful distances.

Efficient representation involves the choice of suitable parameters set such that the objective function $F(\mathbf{x})$ can be factorized as

$$F(x) = \sum_{i,j} \left[\mathbf{S}^T(\mathbf{x}) \mathbf{P}(p_{ij}) \right]^2 = \mathbf{S}^T(\mathbf{x}) \mathbf{M} \mathbf{S}(\mathbf{x}), \tag{9}$$

where **S** and **P** are vector field of the same dimensions and

$$\mathbf{M} = \sum_{i,j} \mathbf{P}(p_{ij}) \mathbf{P}^T(p_{ij}) . \tag{10}$$

The above formulation can be exploited in gradient based methods to save the computational efforts by calculating the matrix **M** in advance. For instance, (3) and (8) are also examples of efficient parametrization. Further computational resources can be saved, if the parametrization is chosen such that **S(x)=x**, which implies the gradient and Hessian matrix of (9) are 2**Mx** and 2**M**, respectively. Nevertheless, this approach might increase the length of the parameters vector [8].

C. Constraints

The enforcement of constraints is required to provide a *meaningful* reconstructed object at the end of the optimization procedure; by meaningful, we intend suitable for industrial purposes, possibly in accordance to the original designer's intent and editable in a CAD environment. The significance of each mechanical part of the reconstructed model is therefore

strongly related to the number, the quality and the validity of the constraints [10], [14]. For instance, in the reconstruction of a cylinder, perpendicularity of the base surfaces with respect to the lateral one is mandatory, even at the expense of a poorer fitting to the acquired data (e.g., if they are strongly corrupted); otherwise, the reconstructed object would be barely useful, failing the main purpose of RE of mechanical parts.

There exist two main approaches for the definition of constraints in the literature: automatic (or semi-automatic) detection [10], [21]–[23] and user defined [9], [15], [24], [25]. Automatic constraints recognition in RE is a very attractive field, because relieves the designer from a tedious and long task and allows less experienced users to perform the reconstruction process. Nevertheless, it still represents a challenging goal when considering very complicated objects.

Whichever the approach to constraints detection is, the validation of constraints represents another key aspect in RE. Indeed, the probability that a human or a machine-guided procedure introduce redundant or contradicting constraints increases as the number of parts of a mechanical object grows. While in the former case an unnecessary complexity has to be faced, in the latter the reconstruction process is likely to fail or never converges; hence, automatic routines capable to manage detect and, eventually, resolve such situations are desired. However, a discussion of such methods is beyond the scope of this paper.

Interestingly, there is no common starting point for several authors whether only equality constraints [8], [10], [21] or inequality constraints [9], [15], [23], [25] should be considered in the RE framework. Inequality constraints are more general, since equalities can be rewritten as two complementary inequalities. Furthermore, inequalities allow to consider tolerances on mechanical parts and seem more suitable from an engineering perspective. However, there are circumstances where reasonable values for tolerances cannot be easily determined, because constrained relations are cumbersomely connected to physical properties of the object (see intrinsic constraints below). Additionally, noticeable optimization routines have been devised in RE field considering equality constraints. Thus, for sake of completeness, both kinds of relations have been expressed in (1).

Constraints can be mainly classified into two categories [9], [15]: intrinsic and extrinsic. Constraints belonging to the former class enforce geometric properties of the shape of a given surface. For instance, given the generic quadric surface equation is

$$f(x,y,z)=\begin{bmatrix}x & y & z\end{bmatrix}\begin{bmatrix}a & h & g\\ h & b & f\\ g & f & c\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}+\begin{bmatrix}u & v & w\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}+d=0, \quad (11)$$

specific conic surfaces can be enforced in the reconstruction process by imposing different mathematical relations on the parameters (please note that (10) can be rearranged in the form of (2)). In the case of a sphere, the constraints in (1) can be written as

$$\begin{aligned}&\mathbf{g}(a,d,u,v,w)=-u^2-v^2-w^2+ad\le 0\\ &\mathbf{c}(a,b,c,f,g,h)=\begin{cases}a-b=0, & b-c=0\\ f=0, & g=0, \quad h=0,\end{cases}\end{aligned} \quad (12)$$

where the inequality assures a non-negative radius (degenerate cases of plane (a=0) and point are allowed). Please note that introducing tolerances in the parameters of (12) would allow a morphing of the sphere; nevertheless, this approach is questionable from a reconstruction that is carried out following a feature-based paradigm, because we started assuming to look for a sphere and not for something that is *like* a sphere. Additionally, poor tolerance settings may cause hardly predictable geometrical results in the reconstructed object.

Extrinsic constraints enforce topological and geometric relations between different surfaces of an object. Parallelism, orthogonality, relative orientations, concentricity, coaxiality, tangency, predetermined distances are some examples of features that are part of this class. For instance, orthogonality between two planes, namely $\mathbf{h_{p1}}$ and $\mathbf{h_{p2}}$, is given by

$$\mathbf{c}(u_1,v_1,w_1,u_2,v_2,w_2)=u_1u_2+v_1v_2+w_1w_2=0\,, \quad (13)$$

A know tolerance angle $\tau \ge 0$ taking into account eventual mechanical or production imperfections can be introduced by modifying (13) and considering inequality relations:

$$\begin{aligned}&\mathbf{g}(u_1,v_1,w_1,u_2,v_2,w_2)=\begin{cases}u_1u_2+v_1v_2+w_1w_2-\cos(\tau)<0\\ -u_1u_2-v_1v_2-w_1w_2-\cos(\tau)<0\end{cases}.\\ &\mathbf{c}(u_1,v_1,w_1,u_2,v_2,w_2)=\begin{cases}u_1^2+v_1^2+w_1^2-1=0\\ u_2^2+v_2^2+w_2^2-1=0\end{cases}\end{aligned} \quad (14)$$

The equality relation in (14) assures unit norm of the parameters; they can be considered as intrinsic constraints and tolerances on them should be avoided. A substantial list of constraints in both 2-D and 3-D spaces is provided in [8].

Although common constraints can be usually described by linear or quadratic form, more complex mathematical descriptions may arise in the formulation of the RE problem. In order to contain the complexity burden, *auxiliary* objects can be introduced [8]. Auxiliary objects are virtual geometric structures, such as points, lines, planes, that are associated with no acquired data, but that are devised only to simplify the constraints formulation. A typical example is the enforcement of the intersection of three or more planes in a point: in this case it is convenient to extend $\mathbf{x}$ to $\mathbf{x^1}=[\mathbf{x}^T, \mathbf{x_a}^T]^T$, introducing the virtual point $\mathbf{x_a}=[x_a,y_a,z_a]^T$ in the parameters, and forcing it to lie on each planes. Auxiliary objects are introduced at the expense of more mathematical relations, but they are not involved in the computation of the objective function.

Finally, it is worth to note that the analytical expression of constraints plays an important role in the classification of the minimization problem (1) and, consequently, in its solvability. This aspect is particularly crucial for optimization procedures based on gradient methods.

III. Gradient Based Optimization Algorithms

The study of optimization techniques based on gradient-based methods is a long-term topic. In this paper, we focus on two main gradient based optimization techniques for constrained fitting in RE field: sequential unconstrained minimization [9] and sequential constraints satisfactions [8]. Even though other remarkable study on RE have been presented more recently, they still represent state-of-art in the field. They require that both the objective function and the constraints relations are continuously differentiable. Moreover, they solve local optimization problems; if a non-convex minimization problem is considered in (1), these methods may only converge to a local minimum of $F(\mathbf{x})$.

A. Sequential unconstrained minimization

In sequential unconstrained minimization, only quadric surfaces and constraints are managed. Specifically, the objective function is strictly convex and is on the form of (5), where positive definiteness of the system matrix **H** is enforced by both its dependence of acquired data and its construction by means of (3) and (4). As to the constraints, they are generally considered by inequalities of the form

$$g_k(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{x} + c \le 0, \quad k = 1,\ldots,M_i, \tag{15}$$

where **A** is not generally positive definite or positive semidefinite. Inequality (15) would generally induce a non-convex problem. Hence, the modified version (16) is used in (1) to assure convexity:

$$\bar{g}_k(\mathbf{x}) = [g_k(\mathbf{x})]^2 \le 0, \quad k = 1,\ldots,M_i. \tag{16}$$

By eventually assuming that all g_k have linearly independent gradient, it is known that (1) is a convex optimization problem and it satisfies the Karun-Kush-Tucker conditions [26]; hence, its unique solution exists and it is also the unique solution of the unconstrained problem (17) for an optimal positive weight vector $\boldsymbol{\lambda}=[\lambda_{1,\ldots,}\lambda_{\mathrm{Mi}}]^T$:

$$\textbf{minimize} \quad F(\mathbf{x}) + \sum_{k=1}^{M_i}\lambda_k \bar{g}_k(\mathbf{x}), \quad \lambda_k > 0,\ k = 1,\ldots,M_i. \tag{17}$$

Since (17) is a convex and continuously differentiable problem, equating its gradient (with respect to both **x** and **λ**) to zero theoretically suffices to compute the optimal set of parameters **x**. Unfortunately, this approach is convenient only when further assumptions on the mathematical structure yield to a closed form solution of (17). In the general case, the values of **λ** are unknown and numerical routines must be used.

Sequential unconstrained minimization is an iterative routine that exploits the convergence of (17) towards the same solution of (1) for increasing values of **λ**. The idea is the following: at the *m*-th step, elements of $\boldsymbol{\lambda}_{(m)}$ are increased from their previous values $\boldsymbol{\lambda}_{(m-1)}$ and are substituted in (17), in order to obtain an unconstrained minimization convex problem; then it is solved with respect to $\mathbf{x}_{(m)}$ starting from the previous iteration's value $\mathbf{x}_{(m-1)}$. The procedure is repeated until all constraints are satisfied. The algorithmic structure is provided below:

1. Initialize $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{(0)}$ and $\mathbf{x} = \mathbf{x}_{(0)}$
2. While $\exists\ k$ such that $\bar{g}_k(\mathbf{x}) > 0$ do
   a. $\boldsymbol{\lambda} = Update(\boldsymbol{\lambda})$
   b. Set $e(\cdot) = F(\cdot) + \sum_{k=1}^{M_i}\lambda_k \bar{g}_k(\cdot)$
   c. $\mathbf{x} = LMsolver[e(\cdot), \mathbf{x}]$
3. End While

At step 1, $\mathbf{x}_{(0)}$ is computed by merging the solutions of the unconstrained fitting problems (3) for each surface, considering only intrinsic constraints (e.g.: normalization constraints, positive radius). Then the *k*-th element of $\boldsymbol{\lambda}_{(0)}$ is obtained according to $\lambda_k = F\left(\mathbf{x}_{(0)}\right)/\bar{g}_k(\mathbf{x}_{(0)})$, in order to avoid the trivial zero vector solution. The initialization of the parameter vector is key point to guarantee the convergence of the algorithm as well as to minimize the number of iterations.

After the updating of elements of **λ,** which is usually carried out by simply adding positive constants, the solution **x** is updated at step 2c by minimizing $e(\mathbf{x})$ according to a Levenberg-Marquardt routine [27]. Specifically, *LMsolver* requires the error function $e(\cdot)$ and the starting point **x** as well two positive quantities that determine the optimization behavior, that is, the initial step s_0 and the on the tolerance $\varepsilon_{\mathrm{th}}$. In the following, the sketch of the *LMSolver* is reported (**I** is the identity matrix):

1. Initialize $s=s_0$, $\varepsilon = \infty$
2. While $\varepsilon > \varepsilon_{\mathrm{th}}$
   a. Set $\varepsilon_0 = e(\mathbf{x})$
   b. Compute $\mathbf{G}$ = gradient[$e(\mathbf{x})$]
   c. Compute $\mathbf{H}$ = Hessian[$e(\mathbf{x})$]
   d. Compute $\mathbf{H_m} = \mathbf{H} + s\mathbf{I}$
   e. Solve $\mathbf{H_m}\,\Delta\mathbf{x} = -\mathbf{G}$ w.r.t. $\Delta\mathbf{x}$
   f. $\varepsilon_{\mathrm{new}} = e(\mathbf{x}+\Delta\mathbf{x})$
   g. If $\varepsilon_{\mathrm{new}} < \varepsilon$
      i. Set $\mathbf{x} = \mathbf{x}+\Delta\mathbf{x}$, $\varepsilon = \varepsilon_{\mathrm{new}}$
      ii. $s = Decrease(s)$
   h. Else
      i. $s = Increase(s)$
      ii. Go to point 2d
   i. End If
3. End While

It has to be pointed out that several variants of such algorithm exist, considering different stopping conditions of Hessian update. Furthermore, the efficiency is strongly related on the implementation of *Decrease*() and *Increase*() functions, as well as on the quantity checked at step 2e. Optimal settings in such sense are strongly dependent on the scale of error function $e(\cdot)$. Werghi has also discussed a numerical instability issue of Levenberg-Marquardt algorithm related on the update of λ, proposing a variant of *LMSolver* to overcome the problem. For sake of space it has been omitted; the interested reader can refer to [9].

B. Sequential constraints satisfactions

Sequential constraints satisfaction is an iterative constrained fitting optimization routine that manages only equality constraints. At each step of the procedure, the validity of the constraints is checked according to specified priorities and the unfeasible ones are discarded; furthermore, efficient exploration of the solution space is performed by using a minimum set of independent parameters. In the original paper [8], a quadratic objective function has been used, since faithful and efficient representations of the surfaces are adopted. Even though this assumption reduces the computational burden, is not generally required for the validity of the method.

The constrained fitting problem in (1) is firstly approximated and reduced to a compact form. We start by considering an approximate version

$$\begin{aligned} &\textbf{minimize} \quad F(\mathbf{x}) \approx F(\bar{\mathbf{x}}) + \mathbf{F}'(\bar{\mathbf{x}})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{F}''(\bar{\mathbf{x}})^T \Delta\mathbf{x} \\ &\textbf{subject to contraints} \quad \mathbf{c}(\mathbf{x}) \approx \mathbf{c}(\bar{\mathbf{x}}) + \mathbf{c}'(\bar{\mathbf{x}})\Delta\mathbf{x} = \mathbf{0}, \end{aligned} \tag{18}$$

where the objective function and the equality constraints have been replaced by their second order and first order Taylor expansions, respectively. In (18) we indicate $\Delta\mathbf{x}=(\mathbf{x} - \bar{\mathbf{x}})$; the gradient and the Hessian matrix of the objective function F as $\mathbf{F}'$ and $\mathbf{F}''$, respectively; moreover, we compact the notation constraints by means of vector $\mathbf{c}$ and matrix $\mathbf{c}'$, that represent the M_e equality constraints functions and their derivatives w.r.t. $\mathbf{x}$, respectively. Without loss of generality, let's assume that the constraint set is ordered according to some given priority scheme, being the most important at the top of vector $\mathbf{c}$. The problem can be finally rewritten by further compacting the notation, defining

$$\mathbf{d} = \begin{bmatrix} \Delta\mathbf{x} \\ 1 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{F}''(\bar{\mathbf{x}}) & \mathbf{F}'(\bar{\mathbf{x}}) \\ \mathbf{F}'(\bar{\mathbf{x}})^T & 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{c}'(\bar{\mathbf{x}}) & \mathbf{c}(\bar{\mathbf{x}}) \end{bmatrix} \tag{19}$$

where $\mathbf{d}$ has length N_d. Substituting (19) in (18), yields

$$\begin{aligned} &\textbf{minimize} \quad \mathbf{d}^T \mathbf{A} \mathbf{d} \\ &\textbf{subject to contraints} \quad \mathbf{C}\mathbf{d} = \mathbf{0}. \end{aligned} \tag{20}$$

The iterative procedure is the following: at the m-th step, the quantities in (19) are computed assuming $\bar{\mathbf{x}} = \mathbf{x}_{(m-1)}$; then a new value of the extended vector $\mathbf{d}_{(m)}$ is obtained by solving (20); finally, the new parameter vector is computed by backward substitution of the first relation in (19), i.e., $\mathbf{x}_{(m)} = \bar{\mathbf{x}} + \mathbf{x}_{(m)}$. The procedure is repeated until some optimality tolerance is satisfied.

The solving procedure for (20) is divided in two phases: dimensionality reduction and unconstrained minimization. The former step aims at extracting a minimum set of independent parameters that satisfy the constraints by means of a Gaussian-elimination-like procedure. Starting from the first row of $\mathbf{C}$ (which corresponds to the highest priority constraint) the coefficient c_l having the highest absolute value is selected, being l its position in the row. We firstly observe that by construction of $\mathbf{C}$ necessarily follows that

$$d_l = \sum_{j=1,\, j\neq l}^{N_d} \left(-\frac{c_j d_j}{c_l} \right). \tag{21}$$

Thus, we rewrite $\mathbf{d}$ as

$$\mathbf{d} = \mathbf{M_1 d_1}, \tag{22}$$

being $\mathbf{M_1}$ and $\mathbf{d_1}$ given in (23). Please note that now both the number of columns of the former and the number of rows of the latter are (N_d-1).

$$\mathbf{M_1} = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ -\frac{c_1}{c_l} & \dots & -\frac{c_{l-1}}{c_l} & -\frac{c_{l+1}}{c_l} & \dots & -\frac{c_{N_d}}{c_l} \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix}, \quad d_1 = \begin{bmatrix} d_1 \\ \vdots \\ d_{l-1} \\ d_{l+1} \\ \vdots \\ 1 \end{bmatrix}. \tag{23}$$

Equation (22) can be plugged in the equality constraints of (20), yielding

$$\mathbf{C}\mathbf{d} = (\mathbf{C}\mathbf{M_1})\mathbf{d_1} = \mathbf{0}, \tag{24}$$

Next, the dimensionality reduction process can be iterated starting from the second row of $(\mathbf{C}\mathbf{M_1})$ and so on, until a minimum dimensional vector $\mathbf{d}^*$ is obtained as in (25):

$$\mathbf{d} = \prod_{\mathbf{j}=1}^{K} \mathbf{M_j d^*}, \quad K \leq M_e. \tag{25}$$

The equality in the above relation strictly holds when all the constraints are linearly independent and not contradicting. A key feature of the algorithm is that eventual exceptions are detected during the dimensionality reduction process. If a row is linearly dependent from the previous ones, it will end to

have all null entries at its turn. In this case the procedure jumps to the next row, discarding the constraint. On the contrary, if a constraint contradicts some of the previous ones, its corresponding row will end to have null entries except the last; thus, the procedure can be either halted or the fault constraint can be discarded, continuing to the next row.

The unconstrained minimization phase is eventually carried out at the end of the dimensionality reduction process. By plugging (25) into the objective function of (20), a classical eigenvalue problem is obtained w.r.t. $\mathbf{d}^*$:

$$\textbf{minimize} \quad \left(\mathbf{d}^*\right)^T \left(\prod_{\mathbf{j}=K}^{1} \mathbf{M}_\mathbf{j}^T \mathbf{A} \prod_{\mathbf{j}=1}^{K} \mathbf{M}_\mathbf{j} \right) \mathbf{d}^* . \tag{26}$$

The problem in (26) can be efficiently solved by consolidated numerical routines. Furthermore, computational and memory resources can be saved in the processing of the system matrix by considering that $\mathbf{M_j}$ is sparse. It should be noted that trivial solutions of (26) are avoided because the last element of $\mathbf{d}^*$ is unit.

The overall procedure is summarized in the following. We consider a stopping threshold on the absolute change of $\mathbf{x}$ for sake of synthesis. As to the initialization vector $\mathbf{x_0}$, it can be obtained by unconstrained fitting of each single surface [8].

1. Initialize $\bar{\mathbf{x}} = \mathbf{x_0}$, $\varepsilon = \infty$
2. While $\varepsilon > \varepsilon_{th}$
   a. Compute the quantities in (19)
   b. Compute ($\mathbf{M_1}$... $\mathbf{M_K}$) by dimensionality reduction applied on $\mathbf{C}$
   c. Compute $\mathbf{d}^*$ by solving (26)
   d. Compute $\mathbf{d}$ by means of (25)
   e. Extract $\mathbf{\Delta x}$ from $\mathbf{d}$ according to (19)
   f. $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Delta x}$
   g. $\varepsilon = |\mathbf{\Delta x}|$
3. End While.

IV. Evolutionary optimization algorithms

Optimization by means of evolutionary strategies relies on iterative exploration of the solution space by means of random editing (*mutations*) of previously explored parameters vectors (*population*). Evolutionary algorithms for constrained fitting in the RE field have been proposed by Robertson et al. [25], [28]; their approach is discussed here.

According to the problem statement in (1), an overall parametrization $\mathbf{x}$ of surfaces extracted from a noisy 3-D dataset is assumed (the RANSAC algorithm for extraction of primitives [29], [30] has been adopted in the original paper). The objective function is based on the Euclidean distance and it is defined as

$$F(\mathbf{x}) = \sum_n \min_{i=1,\ldots,N} \left\| p_n - s_i(\mathbf{x_i}) \right\|^2 \tag{27}$$

where the index n runs on all the points dataset. Noticeably, (27) is not convex in general nor requires a fixed association between each point and a surface. As to the constraints set, it is divided in three groups: linear constraints (both equalities and inequalities), nonlinear equalities and nonlinear inequalities, namely {L}, {Ne} and {Ni}, in that order.

The procedure is based on the GENOCOP III algorithm, which, in turn, is based on GENOCOP I algorithm [31]. For sake of space, only an outline of the GENOCOP III processing is reported here. In GENOCOP III, two populations R and T are considered: the former, namely *reference*, contains only parameters vectors that satisfy all the constraints (*fully feasible*); the latter, *search*, contains vectors parameters that satisfy {L} but not necessarily {Ne} and {Ni}.

At each step, the GENOCOP I algorithm runs over T starting from the best value of the previous iteration $\mathbf{x_{(m-1)}}$, generating a new population T whose elements satisfy necessarily {L}. Then, for each element $\mathbf{t}$ of T that does not fulfil {Ne} or {Ni} a new element $\mathbf{z}$ is randomly generated

$$\mathbf{z} = a\mathbf{t} + (1-a)\mathbf{r} \tag{28}$$

where a is a uniformly distributed random number in (0,1) and $\mathbf{r}$ is an element of R. Equation (28) is repeatedly applied until $\mathbf{z}$ is fully feasible. Then $\mathbf{r}$ is replaced in R by $\mathbf{z}$, whereas $\mathbf{t}$ is replaced in T by $\mathbf{z}$ only with some specified probability. At the end of the step, the best value $\mathbf{x_{(m)}}$ is obtained by evaluating the objective function on all the elements of T and the feasible elements of R.

According to the described procedure, the evolutionary exploration of the space is carried out by the GENOCOP I algorithm at the beginning of each step; however, since it can handle only linear constraints, the subsequent steps of GENOCOP III are necessary to generate new fully feasible parameters for the problem in (1).

An appealing feature of GENOCOP III is that a fully feasible solution is provided at the end of each step, which is not the case of the previous GENOCOP II algorithm [32]. However, initial fully feasible guesses must be provided, whereas this has not required in GENOCOP II.

V. Discussion

In this paper, the constrained fitting problem for RE engineering of mechanical parts has been formalized and discussed. Furthermore, state-of-art techniques devised to solve the problem have been revised, by focusing on the algorithmic structure. The main distinction between two gradient based approaches and an evolutionary one has been drawn not only to distinguish different philosophy solving algorithms, but also because some important practical considerations can be stated.

Firstly, the number of RE methods that rely on gradient based optimization is greater than those ones adopting evolutionary algorithm. A possible explanation has been given in [9]: “We believe that the evolutionary techniques are

suitable mainly to the optimization cases where objective functions and constraints are very complex, presenting hard-handled aspects such nonlinearity, non-differentiability, or do have not explicit forms. Indeed, the earlier mentioned characteristics of these techniques allow them to by-pass these problems. As our optimization problem does not have these problems, the operational research techniques are more appropriate". Nevertheless, to the best of our knowledge, no remarkable work on gradient based optimization implementing the Euclidean distance in 3-D space for has been proposed in the RE field. This aspect may be crucial when considering RE of complex objects starting with low quality data, due to the bias issue discussed in Section II. Moreover, convexity of gradient based methods is also guaranteed by prior points classification, which implicitly relies on a good association of each point with one object's surface. Non-convex objective functions, such as (27), seem more appropriate when this is not the case.

Among the class of gradient optimization, sequential unconstrained minimization represents the first remarkable work. It has been subsequently exploited in a feature-based solid model reconstruction technique working on 2-D contours [23]. A very similar approach has been used for the decomposition of complex RE problems [22]. In [21], it has been extended to handle B-spline for the reconstruction of 3-D surfaces from 2-D sectional curves. Remarkably, the authors of [21] also affirm that sequential unconstrained minimization has been preferred to sequential constraints satisfaction due its superior robust convergence performance.

Sequential constraints satisfaction overcomes some important limitations of sequential unconstrained minimization, such as managing of quadric surfaces, complex constraints formulation, requirement of constraints feasibility and independence. It has been successfully adopted in noticeable RE solutions, such as reconstruction of topologically consistent B-REP models [33], automatic recognition of surface constraints [10], as well as for beautification process [34]–[36]. This last class of approaches firstly performs an independent fitting of surfaces and subsequently enforces geometric relations; any deviation analysis w.r.t. the reference data is in this case omitted, as the only input considered for the imposition of constraints are the previously-fitted surfaces. The decoupling of reference data and constraints allows for a more efficient formulation of the reconstruction problem, but also imposes a simplification that has consequences on the quality of the obtained result.

Evolutionary algorithms have been widely considered in several scientific problems due to their specific attractive features: i) no requirement on convexity (global optimization), ii) no requirement on continuity or differentiability, iii) easy implementations, iiii) they often succeed in practical applications where other methods (e.g., gradient-based optimization) cannot be applied. On the other side, convergence properties are theoretically demonstrated for few simple cases; moreover, routines might be very resource demanding. For the interested reader, Fayolle and Pasko [37] have considered an evolutionary approach for the related problem of reconstructing the feature tree starting from primitives unconstrainedly fitted on an acquired 3-D dataset.

Constrained fitting represents, in general, a valid approach to tackle the reconstruction of mechanical parts and to consider significant geometric constraints in the final result; several variations of the basic approach have been proposed in the scientific literature and also implemented, to some degree, in most advanced reverse engineering software packages (e.g. [38]).

While the models generated with this class of approaches can better represent the original design intent of the designer, it must be underlined that constrained fitting aims at reconstructing the geometrical information of an object; it therefore represents only a part of the entire RE process. By considering recent trends in the field, several efforts have been spent to devise new techniques for both solid features recognition and topology reconstruction. Indeed, topological consistency of the reconstructed object is a fundamental feature to grant subsequent editability by a CAD user. The retrieval of a complete modelling history, provided with associativity relations, defining a parametric representation for the reconstructed CAD model, is still an open issue in the RE field. Additionally, a parametric representation compatible with at least one major renowned CAD/CAE software package is the result desired by reverse engineers;

It must be noted that the retrieval of an object's modeling history is a problem characterized by more than one solution, even in most simple cases. As an example, the modeling of a solid cylinder could be achieved by using one of the following procedures: i) a revolution function (revolving a segment parallel to the revolution axis), ii) an extrusion of a circle or iii) a loft operation with a segment as guide curve and a circle as section. In this example, each representation is equally valid if additional context is not provided. A CAD user could prefer depending on his/her needs, the application, and possible modifications that need to be executed on the model, one solution over the others.

Up to this date the only viable option to assure a satisfying result is to rely on a user-guided reconstruction framework carried out within a dedicated software (e.g. Geomagic Design X, Polyworks). Such systems are provided with a parametric modeling environment that allows the user to select the preferred reverse modeling strategy. Recently, a novel reconstruction strategy, inspired by the constrained fitting approach, has been proposed in [39]; it aims at the reconstruction of parametric CAD models defined by a suitable associative modelling history. The authors are currently studying a possible solution based on the proposed reconstruction paradigm that will be hopefully discussed in a future publication.

REFERENCES

[1] E. Solaberrieta, R. Minguez, L. Barrenetxea, E. Sierra, and O. Etxaniz, "Computer-aided dental prostheses construction using reverse engineering," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, no. 12, pp. 1335–1346, Sep. 2014.

[2] P.-T. DOUTRE *et al.*, "Comparison of some approaches to define a CAD model from topological optimization in design for additive manufacturing," 2017, pp. 233–240.

[3] A. Arles, P. Clerc, G. Sarah, G. Bonnamour, J. Heckes, and A. Klein, "3D reconstruction and modeling of subterranean landscapes in collaborative mining archeology projects: techniques, applications and experiences," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-5/W2, 2013 XXIV International CIPA Symposium*, 2013.

[4] R. Furferi, L. Governi, Y. Volpe, L. Puggelli, N. Vanni, and M. Carfagni, "From 2D to 2.5D i.e. from painting to tactile model," *Graphical Models*, vol. 76, no. 6, pp. 706–723, Nov. 2014.

[5] Y. Volpe, R. Furferi, L. Governi, and G. Tennirelli, "Computer-based methodologies for semi-automatic 3D model generation from paintings," *International Journal of Computer Aided Engineering and Technology*, vol. 6, no. 1, p. 88, 2014.

[6] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-Aided Design*, vol. 29, no. 4, pp. 255–268, 1997.

[7] L. Governi, R. Furferi, M. Palai, and Y. Volpe, "3D geometry reconstruction from orthographic views: A method based on 3D image processing and data fitting," *Computers in Industry*, vol. 64, no. 9, pp. 1290–1300, Dec. 2013.

[8] P. Benko, G. Kós, T. Várady, L. Andor, and R. Martin, "Constrained fitting in reverse engineering," *Computer Aided Geometric Design*, vol. 19, no. 3, pp. 173–205, 2002.

[9] N. Werghi, R. Fisher, C. Robertson, and A. Ashbrook, "Object reconstruction by incorporating geometric constraints in reverse engineering," *CAD Computer Aided Design*, vol. 31, no. 6, pp. 363–399, 1999.

[10] I. Kovács, T. Várady, and P. Salvi, "Applying geometric constraints for perfecting CAD models in reverse engineering," *Graphical Models*, vol. 82, pp. 44–57, 2015.

[11] R. B. Fisher, "Applying knowledge to reverse engineering problems," *Computer-Aided Design*, vol. 36, no. 6, pp. 501–510, May 2004.

[12] L. Di Angelo and P. Di Stefano, "Geometric segmentation of 3D scanned surfaces," *Computer-Aided Design*, vol. 62, pp. 44–56, 2014.

[13] A. Shamir, "A survey on mesh segmentation techniques," *Computer Graphics Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.

[14] A. I. Protopsaltis and I. Fudos, "A feature-based approach to re-engineering CAD models from cross sections," *Computer-Aided Design and Applications*, vol. 7, no. 5, pp. 739–757, 2010.

[15] N. Werghi, R. Fisher, A. Ashbrook, and C. Robertson, "Shape reconstruction incorporating multiple nonlinear geometric constraints," *Constraints*, vol. 7, no. 2, pp. 117–149, 2002.

[16] A. Lapini, F. Argenti, A. Piva, and L. Bencini, "Comparison of super-resolution methods for quality enhancement of digital biomedical images," in *2014 8th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2014, pp. 1–5.

[17] P. Faber and R. B. Fisher, "Euclidean Fitting Revisited," Springer, Berlin, Heidelberg, 2001, pp. 165–175.

[18] P. Faber, P. Faber, and B. Fisher, "A Buyer's Guide to Euclidean Elliptical Cylindrical and Conical Surface Fitting," *PROC. BRITISH MACHINE VISION CONFERENCE BMVC01*, pp. 521--530, 2001.

[19] G. Taubin and Gabriel, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[20] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[21] Y. Ke, S. Fan, W. Zhu, A. Li, F. Liu, and X. Shi, "Feature-based reverse modeling strategies," *CAD Computer Aided Design*, vol. 38, no. 5, pp. 485–506, 2006.

[22] A. Karniel, Y. Belsky, and Y. Reich, "Decomposing the problem of constrained surface fitting in reverse engineering," *CAD Computer Aided Design*, vol. 37, no. 4, pp. 399–417, 2005.

[23] J. Wang, D. Gu, Z. Gao, Z. Yu, C. Tan, and L. Zhou, "Feature-Based Solid Model Reconstruction," *Journal of Computing and Information Science in Engineering*, vol. 13, no. 1, p. 11004, 2013.

[24] R. B. Fisher, "Applying knowledge to reverse engineering problems," *Proceedings - Geometric Modeling and Processing: Theory and Applications, GMP 2002*, vol. 36, pp. 149–155, 2002.

[25] C. Robertson, R. Fisher, N. Werghi, and A. P. Ashbrook, "Fitting of constrained feature models to poor 3d data," *Evolutionary Design and Manufacture*, pp. 149–160, 2000.

[26] A. W. Kuhn, H. W.; Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1950, pp. 481–492.

[27] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[28] C. Robertson, R. B. Fisher, D. Corne, N. Werghi, and A. Ashbrook, "Investigating Evolutionary Optimisation of Constrained Functions to Capture Shape Descriptions from Range Data," in *Advances in Soft Computing*, London: Springer London, 1999, pp. 455–466.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[30] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[31] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, vol. 24, no. 3. Springer-Verlag, 1997.

[32] Z. Michalewicz and G. Nazhiyath, "Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints," in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, 1995, vol. 2, pp. 647–651.

[33] P. Benko, R. R. Martin, and T. Várady, "Algorithms for reverse engineering boundary representation models," *CAD Computer Aided Design*, vol. 33, no. 11, pp. 839–851, 2001.

[34] F. C. Langbein, A. D. Marshall, and R. R. Martin, "Choosing consistent constraints for beautification of reverse engineered geometric models," *Computer-Aided Design*, vol. 36, no. 3, pp. 261–278, 2004.

[35] M. Li, F. C. Langbein, and R. R. Martin, "Detecting design intent in approximate CAD models using symmetry," *CAD Computer Aided Design*, vol. 42, no. 3, pp. 183–201, 2010.

[36] M. Li, F. C. Langbein, and R. R. Martin, "Detecting approximate symmetries of discrete point subsets," *Computer-Aided Design*, vol. 40, no. 1, pp. 76–93, Jan. 2008.

[37] P.-A. Fayolle and A. Pasko, "An evolutionary approach to the extraction of object construction trees from 3D point clouds," *Computer-Aided Design*, vol. 74, pp. 1–17, May 2016.

[38] 3D Systems, "Geomagic Design X (formerly Rapidform XOR)," 2016. [Online]. Available: http://www.rapidform.com/products/xor/overview/. [Accessed: 29-Sep-2016].

[39] F. Buonamici and M. Carfagni, "Reverse Engineering of Mechanical Parts: A Brief Overview of Existing Approaches and Possible New Strategies," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016, p. V01BT02A003.