Journal of Computational Design and Engineering 5 (2018) 145–159

Contents lists available at ScienceDirect

Journal of Computational Design and Engineering

journal homepage: www.elsevier.com/locate/jcde

Reverse engineering of mechanical parts: A template-based approach

Francesco Buonamici *, Monica Carfagni, Rocco Furferi, Lapo Governi, Alessandro Lapini, Yary Volpe

Department of Industrial Engineering, University of Florence, Via di Santa Marta 3, 50139 Florence, Italy

ARTICLE INFO

Article history:
Received 14 July 2017
Received in revised form 10 November 2017
Accepted 10 November 2017
Available online 14 November 2017

Keywords:
Reverse engineering
CAD reconstruction
CAD template
3D mesh
Constrained fitting
Particle Swarm Optimization

ABSTRACT

Template-Based reverse engineering approaches represent a relatively poorly explored strategy in the field of CAD reconstruction from polygonal models. Inspired by recent works suggesting the possibility/opportunity of exploiting a parametric description (i.e. CAD template) of the object to be reconstructed in order to retrieve a meaningful digital representation, a novel reverse engineering approach for the reconstruction of CAD models starting from 3D mesh data is proposed. The reconstruction process is performed relying on a CAD template, whose feature tree and geometric constraints are defined according to the a priori information on the physical object. The CAD template is fitted upon the mesh data, optimizing its dimensional parameters and positioning/orientation by means of a particle swarm optimization algorithm. As a result, a parametric CAD model that perfectly fulfils the imposed geometric relations is produced and a feature tree, defining an associative modelling history, is available to the reverse engineer. The proposed implementation exploits a cooperation between a CAD software package (Siemens NX) and a numerical software environment (MATLAB). Five reconstruction tests, covering both synthetic and real-scanned mesh data, are presented and discussed in the manuscript; the results are finally compared with models generated by state of the art reverse engineering software and key aspects to be addressed in future work are hinted at.

1. Introduction

The reconstruction of digital geometric models of physical objects, usually indicated as Reverse Engineering (RE) in the Computer Aided Design (CAD) field, has been extensively studied in recent years, due to the development and spreading of 3D scanning technologies and the increase in number of potential applications (Burston, Sabatini, Gardi, & Clothier, 2014; Solaberrieta, Minguez, Barrenetxea, Sierra, & Etxaniz, 2014; Voicu, Gheorghe, Badita, & Cirstoiu, 2013). Most advanced processes, in fact, exploit 3D data acquired on the physical object and describing its surfaces as starting point for the reconstruction framework (Chang & Chen, 2011; Várady, Martin, & Cox, 1997) (Fig. 1).

Considering engineering applications, the main goal of the RE process is to extract information from the acquired raw data to reconstruct a proper parametric CAD model that is as close as possible to the original design of the object. The composing CAD features are specifically required to be correct in dimensions, combinatorial structure and in the existing relations (i.e. geometric constraints, symmetries, regularities) between them.

The practical usefulness of the model obtained at the end of the RE process depends on multiple factors, the most important being the ability to recover the intrinsic design intent defining the part. The achievement of a "close representation" is, in fact, usually not sufficient for engineers, because a model defined with the correct set of geometric relations and properties, as well as the correct dimensions, is needed.

The correct functioning of mechanical parts, as an example, 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 (Benko, Kós, Várady, Andor, & Martin, 2002).

The state of the art of RE approaches is wide and assorted, as proved by multiple surveys recently presented at the state of the art (Anwer & Mathieu, 2016; Buonamici & Carfagni, 2016; Chang & Chen, 2011; Geng & Bidanda, 2017); nevertheless, fundamentally two approaches have been proposed to achieve the previously described goal by integrating geometric constraints in the reconstruction: *Constrained Fitting* (Benko et al., 2002; Werghi, Fisher, Robertson, & Ashbrook, 1999) and *Beautification* (Gao, Langbein,

Abbreviations: RE, Reverse Engineering; CAD, Computer Aided Design; CAE, Computer aided engineering; TCRT, Template-Based CAD Reconstruction Tool; PSO, Particle Swarm Optimization; CSG, Constructive Solid Geometry; COM, Component Object Model.

Peer review under responsibility of Society for Computational Design and Engineering.

* Corresponding author.
E-mail addresses: francesco.buonamici@unifi.it (F. Buonamici), monica.carfagni@unifi.it (M. Carfagni), rocco.furferi@unifi.it (R. Furferi), lapo.governi@unifi.it (L. Governi), alessandro.lapini@unifi.it (A. Lapini), yary.volpe@unifi.it (Y. Volpe).

https://doi.org/10.1016/j.jcde.2017.11.009
2288-4300/

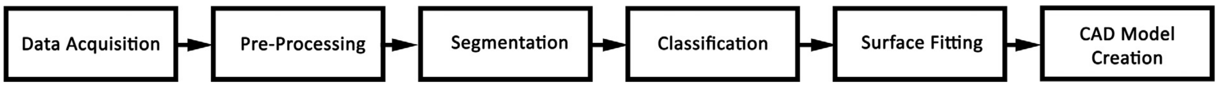


Fig. 1. Traditional RE framework.

Marshall, & Martin, 2004; Langbein, Marshall, & Martin, 2004; Langbein, Mills, Marshall, & Martin, 2001).

Constrained fitting methods (Benko et al., 2002; Werghi et al., 1999) pursue a more meaningful reconstruction introducing a set of geometrical constraints that are defined among the identified geometric features and enforced during the "Surface Fitting" step (Fig. 1) by formulating a constrained optimization problem. Satisfactory results are achievable with this technique, but a considerable effort is required to obtain a suitable set of constraints, either by using an automatic recognition algorithm or relying on the user contribution. The definition of a solvable non-contradictory and convenient constraint set is a non-trivial task that greatly influences the efficiency of the method. In most *constrained fitting* formulations, furthermore, the constraints are implemented only up to a certain tolerance (Buonamici & Carfagni, 2016). Even though the entailed reconstruction errors are usually dimensionally negligible for practical purposes, such imprecisions may be inconvenient in the subsequent editing of the obtained model.

Beautification-based approaches, instead, perform an independent fitting of surfaces to build a first-attempt model. A set of constraints and relations, directly inferred from the reconstructed model, is then enforced as a post-process by changing the surface parameters values; during this phase the original 3D data is ignored, greatly reducing the mathematical complexity of the procedure but also limiting its efficacy (Wang et al., 2013).

Other important aspects that need to be considered to evaluate the performances of a RE method are the type of model generated at the end of the reconstruction process and its usability (Au & Yuen, 1999). Generally, a parametric representation compatible with at least one major renowned CAD/CAE software package is the result desired by reverse engineers; moreover, a significant feature tree and a particular file format may be required: these are usually obtained by means of an intense post-processing phase.

In this context, possible improvements for RE of mechanical parts may be obtainable applying a *knowledge-based* paradigm. As explained in Fisher (2002), the shapes of common mechanical parts "follow standard conventions arising from tradition, utility or engineering design"; accordingly, it may be convenient to spend the prior information about the object at the start of the process. The designer, in fact, has a deep knowledge about the part, which goes beyond the mere geometric analysis that may be carried out on the acquired data by automatic algorithms. In fact, several information sources and clues are available to the reverse engineer and may guide the reconstruction of a correct CAD model: for what purpose was the part designed in the first place, what was its production process (Durupt, Remy, Ducellier, & Pouille, 2014), how it needs to interact with other parts. This information can help the reverse engineer defining the correct topology of the CAD model, its composing features and even to identify geometric constraints and exact dimensions.

One of the most effective knowledge-based approaches is the *template-based* reconstruction (Bosché, 2010; Erdős, Nakano, & Váncza, 2014; Fayolle & Pasko, 2015; Robertson, Fisher, Werghi, & Ashbrook, 2000); in this class of techniques, some kind of CAD template, containing known features and geometric relations, is defined at the start of the RE process and used to guide the reconstruction. The template provides an archetypal description of the shape that has to be found within the acquired data, by imposing a priori desired restrictions in the search process.

In (Robertson et al., 2000), a template-based reconstruction is used to deal with poor quality 3D data pre-processed using a RANSAC approach (Fischler & Bolles, 1981); a series of models which are known in topology (i.e. cylindrical slotted part, square slotted part, etc.), defined with *a priori* known geometric constraints, are sought and fitted by means of a genetic algorithm (Michalewicz, 1997) to the raw data. Satisfying results are shown when dealing with noisy data that would be otherwise difficult to analyse applying traditional techniques; the reconstructed models, however, are still distinguished by loosely-imposed constraints, with a tolerance controlling the constraints strictness.

Analogously, a genetic algorithm (Holland, 1995) is used in Fayolle and Pasko (2015) by Fayolle and Pasko, to perform a reconstruction process based on CAD models generated by CSG (Constructive Solid Geometry). A CAD model obtained with a first independent fitting of identified surfaces is used to start the reconstruction on different point clouds: the values of a set of parameters controlling the model dimensions are obtained by minimizing a least square error function.

Template-based reconstruction has been also studied and applied for civil engineering purposes, such as the tracking of the status of buildings and industrial facilities (Bey, Chaine, Marc, Thibault, & Akkouche, 2012; Bosché, 2010; Erdős et al., 2014). In Bey et al. (2012), the authors focus on the reconstruction of cylindrical parts by means of a probabilistic approach: the template is used to guide the identification of piping elements within the acquired data. The designer knowledge is exploited to assure the consistency of the reconstructed model, such as the correctness of the piping system, which is required to respect a priori known constraints. In Erdős et al. (2014) the method is extended to torus and cuboids.

To the best of authors' knowledge, the results obtained by means of state-of-art template-based RE methods still do not guarantee both the following requirements for usability: (1) the retrieval of a completely defined feature tree associated with a fully editable parametric CAD model; (2) the enforcement of perfect geometric constraints, respecting the user intent and retaining the editability of the model. Starting from these premises, we propose a RE method suitable for the reconstruction of mechanical parts that is well-integrated with the traditional product development process, exploiting tools that are already part of the "engineering culture" of designers (e.g. parametric CAD modelling software environment).

In the present work, a new approach to RE involving the fitting of *a priori* defined CAD template to 3D scanned data is presented (Fig. 2). The template includes the *a priori* information on the part to be reconstructed, that is, the feature tree of the object, the geometrical relations between the features and their known dimensions.

The process is performed within a well-established CAD environment (i.e. Siemens NX) thanks to an appositely devised software tool (named *TCRT* – Template-Based CAD Reconstruction Tool) allowing: (i) a known and familiar environment for the designer, provided with state-of-the-art CAD modelling tools to be used in the template design and (ii) the achievement of a fully parametric and editable model at the end of the reconstruction,

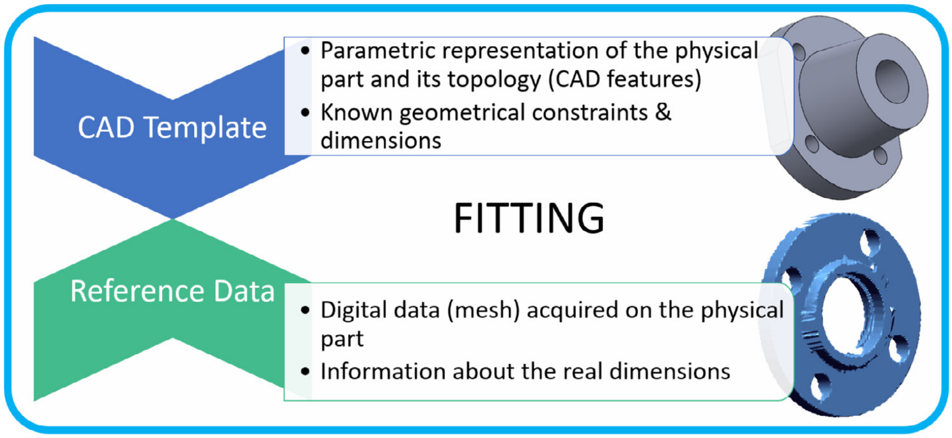


Fig. 2. Fundamental elements of a template-based reconstruction approach.

which is defined with an editable and meaningful associative feature-tree, ready to be used for downstream applications.

Specifically, an optimization process of the parameters controlling the shape, the position and the orientation of the template is carried out to fit the CAD model on the scanned data. The optimization jointly updates the parameters of the whole CAD template (hence implicitly exploiting the pre-determined geometric constraints), intervening directly on the CAD feature tree and minimizing the global fitting error.

The objective function guiding the optimization is built considering deviation data evaluated between the CAD model and the original mesh, using built-in tools of Siemens NX. The proposed TCRT runs on an hybrid MATLAB-Siemens NX implementation to exploit MATLAB proprietary "*Global Optimization Toolbox*": a meta-heuristic optimization algorithm (I.e. Particle-Swarm Optimization – PSO (Kennedy & Eberhart, n.d)) is responsible for the optimization/fitting step.

The proposed method, as well as the algorithm controlling the entire process, are presented in Section 2; results obtained in a series of test cases are discussed in Section 3. Finally, strengths and weaknesses, along with plausible application scenarios, are described in Section 4; possible future improvements are discussed as well.

2. Material and methods

2.1. Overview

A CAD model is completely defined by its modelling history (i.e. the list of modelling functions chronologically ordered in the feature tree) and the parameters of its features (i.e. its dimensions or numerical values in general). Each feature is, in fact, characterized by means of one or more parameters, which account for physical dimensions, e.g. length, angle, number of elements in an array.

We propose a RE approach that aims to fit the CAD template on mesh data by means of an iterative optimization procedure; such routine aims at finding the best CAD features parameters that minimize the global fitting error w.r.t. the acquired data. The method exploits CAD software functionalities such as associative-parametric modelling, enforcing of geometrical constraints and (possibly) evaluating distances between mesh/CAD surfaces. Here and in the following, we are assuming that the feature tree of the CAD template is fixed and suited to accomplish the task (i.e. topologically coherent w.r.t. the physical object). The introduction or the removal of CAD features in the optimization process is, at the moment, not considered.

The main advantage of using a parametric representation based on fixed feature tree inside the CAD environment is the implicit adoption of a complete and meaningful set of geometrical constraints that are always enforced during the optimization process and, consequently, in the final CAD template. The implementation of modelling functions, as well as the compliance of constraints, is transparent to the optimization routine and to the user because it is completely demanded to the CAD software.

As mentioned in introductory section, the proposed RE method has been implemented by using MATLAB and Siemens NX for the optimization routine and CAD environment, respectively. The former provides a ready-to-use developing environment and is supplied with the Global Optimization Toolbox and the Optimization Toolbox, which grant a fast and reliable access to several well-established optimization routines. The latter is a CAD software package equipped by language-neutral API to provide a complete access to CAD core application functionality; furthermore, it supports operations between mesh data and CAD features, such as the computation of distance between a mesh surface and a CAD surface. The interaction between MATLAB and NX has been realized by means of Microsoft .NET Framework and Component Object Model (COM). The principles exposed in this article could be generally extended to different software packages, provided that sufficient programming tools are available[1].

2.2. Implementation

At the beginning of the procedure, the user is required to provide the following inputs:

(1) the initial CAD template representing the object that will be fitted to the acquired data; this object is equipped of both a feature tree and a parameters list defining the desired object topology; in this implementation, it is stored in a NX proprietary format file and it will be referred to as *CAD template*.

Several origins can be imagined for the *CAD template:* in a possible framework, it could be designed specifically for the considered reconstruction by the reverse engineer using traditional NX

[1] Specifically, the optimization routine could be developed in any major programming language of choice (e.g. C++, Python, Visual Basic), provided that a viable interface to the CAD software of choice is available. The choice of a different CAD software, is more difficult as it must allow: (1) a parametric and associative modelling environment; (2) the ability to handle mesh data and perform distance measurement between mesh and CAD surfaces; (3) programming access to the CAD tools. Any CAD software that guarantees these three requirements is a good candidate for replicating the results described in this paper.

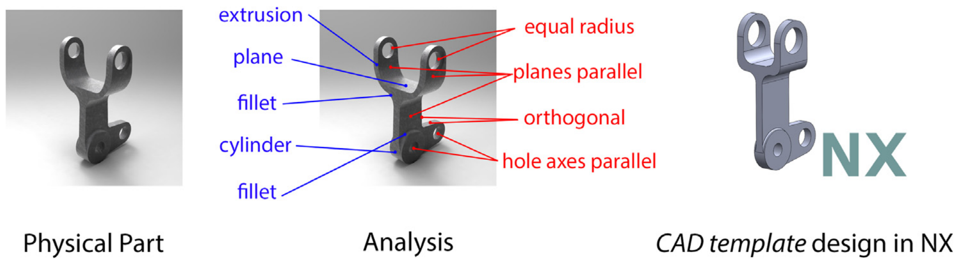


Fig. 3. Possible *CAD template* generation process. In this hypothesis, the reverse engineer analyses the physical part, identifying possible geometric features (coloured in blue) and geometric constraints (coloured in red) to be used for the modelling of the template. The template has a definite topology, defined by the choice made during the modelling of the object (carried out using Siemens NX CAD tools) but does not need to be correct in dimensions or proportions. The *CAD template* in the example required 9 min to be modelled in Siemens NX.

modelling tools. In this hypothesis, the user should first carefully consider the shape of the part to be reconstructed and the possible geometric features and constraints that could have been used and imposed in the original design of the part. The user's engineering skills, the purpose of the part, its functioning and relations, knowledge on the production process used to made it, should all contribute to a correct identification/interpretation of the geometry of the part. Exploiting this information, the user would design a possible CAD representation of the part within the Siemens NX environment, using traditional direct modelling tools, in a process familiar to him/her. It must be noted that the *CAD template* provided does not need to be characterized by dimensions or even proportions between the CAD features that are similar to the physical part ones; the algorithm is capable of handling every possible starting configuration. This greatly simplifies the designing process for the *CAD template*, as the user only needs to pay attention to the correct choice of geometric features and the imposition of the desired constraints; in Fig. 3 is represented an example of the CAD template creation process beforehand described. As previously mentioned, different scenarios for the *CAD template* origin could be considered as well and are discussed in Section 4.

(2) the acquired manifold mesh saved in an STL file, which will be referred to as *manifold STL*;
(3) *N* meshes representing *N* single features obtained by means of a suitable and topologically correct segmentation process of the manifold STL. In this case, a commercial RE software (i.e. RapidWorks: Reverse Engineering Software, n.d, a NextEngine® proprietary version of Geomagic Design X (3D Systems, 2016)) has been used to obtain a significant segmentation, but other methods and tools could be exploited in order to automate the process and to reduce the human interaction required.

Ideally, every segmented region should correspond to a single surface of the *CAD template*; these objects are saved in *N* separated STL files and will be referred to as *STL objects*. In order to achieve this goal, the segmentation could be guided or manually adapted to adhere to the division established by the set of surfaces composing the *CAD template;* the modelling history of the CAD template represents, once again, the reverse engineer's interpretation of the physical part topology; therefore, it is important that the regions identified during the segmentation are in accordance to the reverse engineer' knowledge. Significant discrepancies between the set of CAD surfaces and the regions identified by using a semi-automatic segmentation tool could be taken as hints for an inappropriate modelling of the *CAD template* (i.e. a forced representation of the physical object that results wrong after the segmentation).

The output of the RE process is the *CAD template* updated with the set of optimal parameters, representing the actual shape and dimensions of the physical part. From a logical point of view, the procedure is divided in two stages: a preliminary phase and an optimization phase, hereby described.

2.2.1. Preliminary phase

The goals of the preliminary phase are: (1) registering the *N STL objects* w.r.t. the *CAD template* model; (2) matching (associate) each STL object to the surfaces of the *CAD template*; (3) provide an initial set of CAD parameters. A block diagram of the processing flow is depicted in Fig. 4, where the main steps have been highlighted and displaced in the corresponding application domain. The preliminary phase is started in an interactive session of the NX application domain; the interaction with MATLAB is performed by means of the COM interface.

Registration, though not strictly required, allows to improve the performance of the subsequent optimization phase; in our implementation, a rigid transform that maps the *manifold STL* on the initial CAD model has been obtained by matching centroids and principal axes of inertia of the two models. This is a well-known basic approach (a more refined version based on the same principle, is proposed in Ohbuchi, Otagiri, Ibato, and Takei (2002)) that guarantees a good trade-off between obtained results and simplicity of implementation. Initially, a rotation matrix that aligns the principal axes of inertia of the *manifold STL* w.r.t. the principal axes of the initial CAD *template* is estimated and applied to the *manifold STL* itself.[2] Then a translation vector is estimated to align the centroids of both the *manifold STL* and the initial *CAD template*. Finally, the overall estimated rigid transform (rotation and translation) is applied to the *N STL objects*, obtaining the *N registered STL objects* that are separately saved in *N* corresponding files.

The matching process covers a fundamental role in the preprocessing phase, because it determines which surfaces of the *CAD template* will be considered when evaluating the distance between the *n*-th STL object and the *CAD template*. This association will be kept constant during the entire RE process; it generally follows the *many-to-many* paradigm, that is, the *n*-th STL object can be bind to zero, one or more CAD surfaces, and vice versa. The matching process relies on human interaction, demanding the user to select the appropriate CAD surfaces for each STL object. At the end of this phase, a text file reporting the list of all the N STL object and their associated CAD surface is produced and saved.

In the preliminary phase, a parameters list related to the features of the *CAD template* that will be optimized must be also returned and made available to MATLAB. Furthermore, in order to take into account possible rigid motion of the *CAD template*, parameters related to rotation and translation have to be included

[2] Since this procedure is not capable to resolve three degrees of freedom (i.e. a rotation of 180 degrees for each axis), the user interaction in the NX domain is required to fix the ambiguity and to correctly finalize the estimation of the rotation matrix.

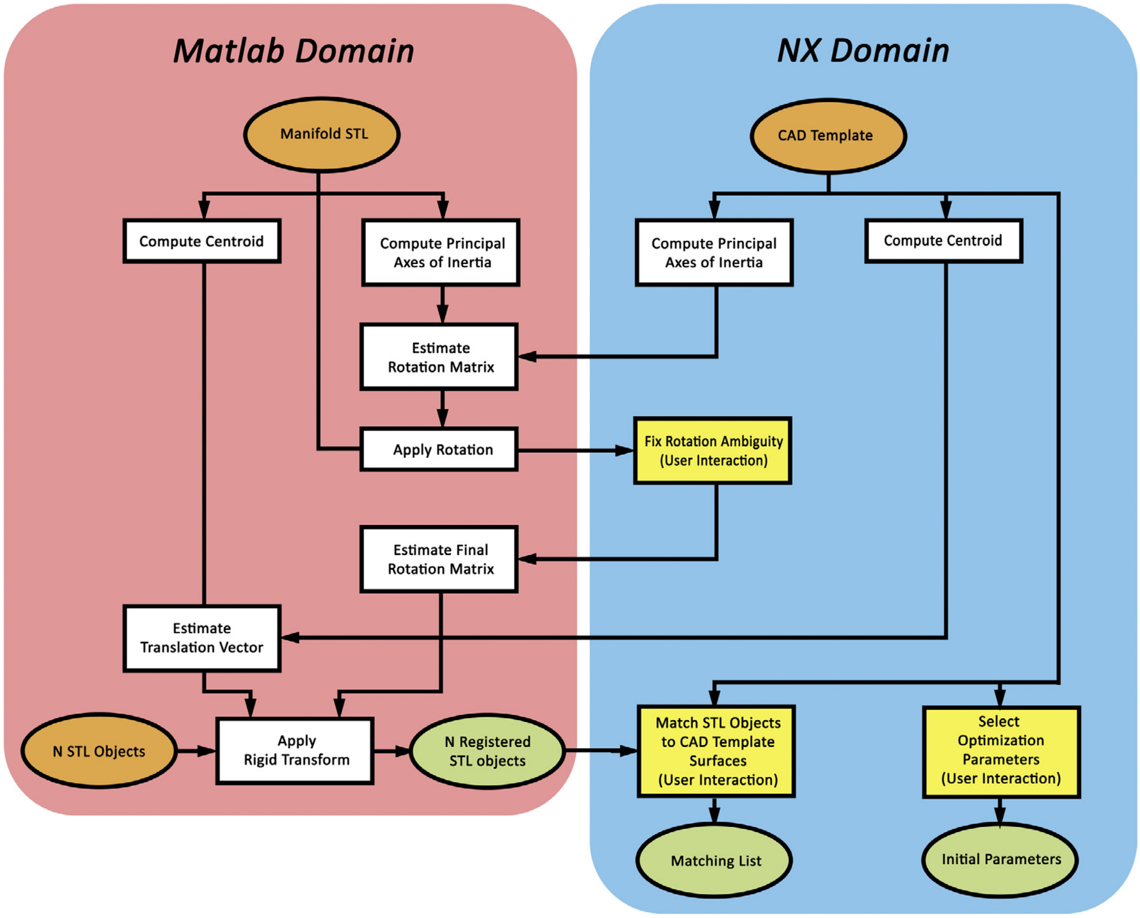


Fig. 4. Processing flow (top-to-bottom) of the preliminary phase, displaced in the correspondent application domains. Orange rounded blocks indicate the input files; white rectangular blocks represent automatic processing, while yellow ones require user interaction; the output files of the procedure are reported in green rounded blocks.

as well. According to NX functionalities, a complete list of all parameters related to the *CAD template* model can be edited in the application domain and exported in a text file. Since only a subset of the complete list usually has to be considered in the optimization phase, the user interaction is required to mark the parameters of interest with a special string flag. On the other hand, unflagged parameters will be kept constant during the remaining procedure.

At the end of the preliminary phase the set of *N registered STL objects*, the matching scheme and the initial set of CAD parameters are obtained. Even though a considerable interaction of the user is demanded, the requested actions do not require particular expertise nor a deep knowledge of the specific CAD environment. Furthermore, the preliminary phase has to be repeated only if the initial *CAD template* or the acquired data or the *N STL objects* change.

2.2.2. Optimization phase

The optimization phase strictly aims at finding the best parameters such that the *CAD template* fits the *N registered STL objects* according to an optimality criterion. With this respect, the minimization of mean Euclidean distance (or mean square error) represents a straightforward choice. In this case, the objective function is defined as follows:

$$F_e = \frac{1}{N}\sum_{n=1}^{N} d_n c \qquad (1)$$

where d_n is the average distance of the n-th STL from the corresponding CAD surfaces, that is:

$$d_n = \frac{1}{P_n}\sum_{p=1}^{P_n} d_{n,p} \qquad (2)$$

being $d_{n,p}$ the Euclidean distance between the p-th vertex of the n-th STL object and the corresponding CAD surfaces, and P_n the number of vertexes of the n-th STL object. Unfortunately, due to the complex relations between parameters and the actual surfaces of the *CAD template*, the analytical relation between the CAD parameters and the objective function is often cumbersome and F_e generally belongs to the class of the non-convex functionals (Locatelli & Schoen, 2013). In these cases, solutions based on local minimization could result inappropriate; thus, global strategies have to be relied on. In this paper, we relied on the *Particle Swarm Optimization* algorithm (Kennedy & Eberhart, n.d) that belongs to the class of Meta-heuristic algorithms, whose adoption is widespread in practical applications (and already successfully applied to data-fitting problems as in Gálvez and Iglesias (2012) and Gálvez and Iglesias (2011)) due to the lower computational burden and lack of strict usage hypothesis w.r.t. exact methods, although the convergence to a global minimum cannot be guaranteed. Indeed, we observed that in considered scenarios the particle swarm algorithm is often misled by local minima when considering the metric of Eq. (2) in the initial period of the optimization routine; in such case an early exit of the procedure occurs, providing a final *CAD template* noticeably “far” from the target STL.

As an alternative to Eq. (2), we empirically found that, in the early stage of the optimization, the following metric is preferable:

$$d_n = \max_{1 \leqslant p \leqslant P_n} d_{n,p} \tag{3}$$

By comparing Eqs. (2) and (3), the substitution of the average with max operator enforces a strict condition on the fitting error of the single STL object. A similar idea could have been used also in Eq. (1), whose minimization would have yielded to a complete *MinMax* approach; however, further tests have shown that the average operator is preferred to avoid an excessive insensitivity of the solver with respect to the optimization parameters. Moreover, it has to be noted that, mathematically, the minimization of Eq. (1) when Eq. (3) is used instead of Eq. (2) generally leads to a different solution; in practice, we observed that a bias in the final CAD model is introduced when dealing with *STL objects* affected by zero-mean stochastic noise. It must be noted that Eq. (3) was adopted over the *Hausdorff* distance (as in Hosseini and Moetakef-Imani (2017)) because the corresponding points (w.r.t. to the STL vertices) on the CAD surfaces are not explicitly available, and a possible evaluation of the *Hausdorff* distance between the two sets would have been more computationally costly.

In accordance to the previous considerations, the optimization procedure has been divided in two stages, in a coarse-to-fine paradigm. In the first stage, the minimization is carried out considering Eq. (1) and the metric in Eq. (3). The parameters of the resulting *CAD template* are then used as starting point for the second stage, where Eq. (2) is considered as metric of Eq. (1). Hence, in the following description we will refer to the optimization phase by generally considering any of the two stages.

The block diagram of the whole optimization phase is summarized in Fig. 5. The user is required to provide the initial parameters file and the specific algorithm settings (which will be discussed in Section 3) in the MATLAB application domain, whereas the *CAD template*, the *N registered STL objects* and the matching list have to be given as initial input in the NX application domain. The entire procedure is led by MATLAB's particle swarm algorithm without user's interaction. Interfaces to the NX computing functionalities are provided through the *.NET* framework, according to a client-server paradigm: at the beginning of each iteration, a new swarm matrix is generated in MATLAB and, for each particle, the objective function is evaluated relying on the NX processing features to compute the distances between *STL objects* and corresponding CAD surfaces.

Specific features of the implemented method are discussed in the following:

(1) *Initialization of the swarm (MATLAB)*. The parameters of the initial *CAD template* are used as seed to randomly generate the initial swarm's particles. For each particle, the value of an optimization parameter is obtained as a realization of a random variable uniformly distributed in a specific numeric interval; additionally, such interval is also enforced during all the optimization phase, that is, each newly generated particles is enforced to fulfill these bounds. Even though parameters space does not theoretically need to be bounded, the definition of upper and lower bounds prevents the particle swarm algorithm to explore useless or redundant regions. In the proposed method, metric parameters (e.g. lengths, diameters, etc.) are bounded between 0 and the main diagonal's length of the *manifold STL*'s bounding box; angular parameters are free to move in the entire round angle. Finally, rigid transform parameters are bounded.

(2) *Passing parameters from MATLAB to NX*. Swarm particles are reported on a specific NX parameters files using the initial parameters file as template. Specifically, a new file is created

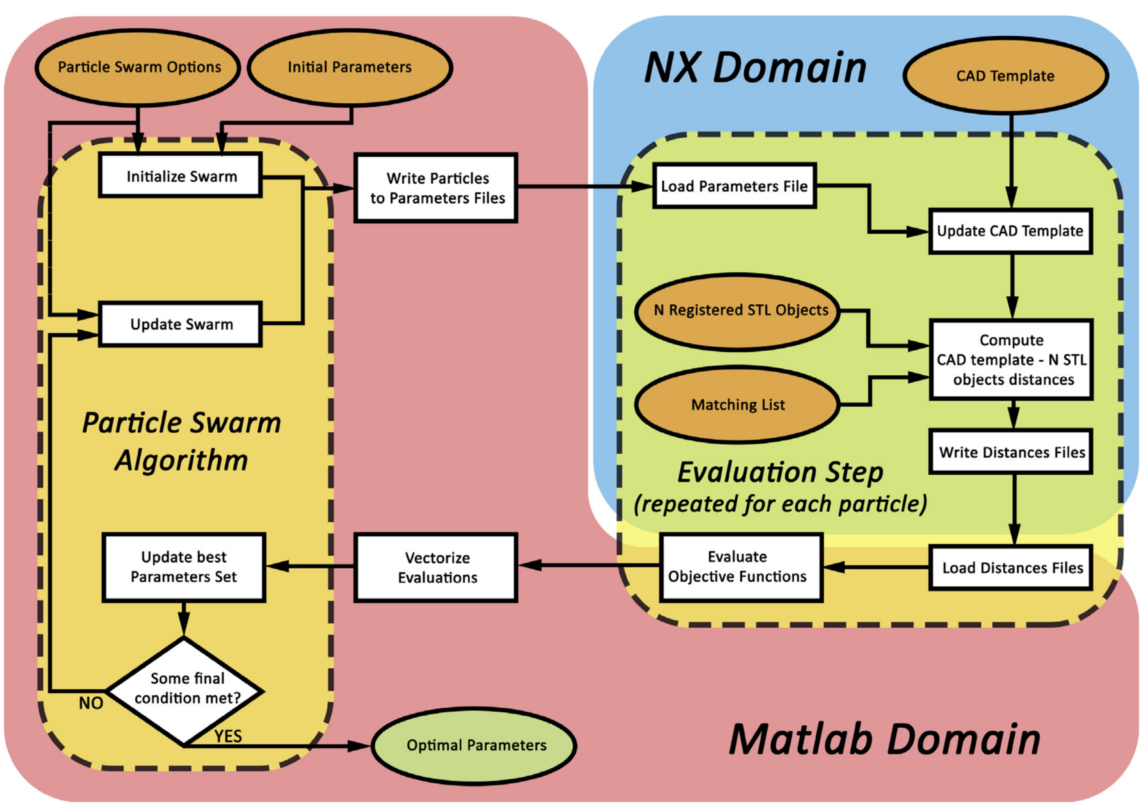


Fig. 5. Processing flow (top-to-bottom) of the optimization phase, in the correspondent application domains. Orange rounded blocks indicate the inputs; white rectangular blocks represent automatic processing; the diamond block represents a conditional evaluation. The optimal parameters are provided at the output of the procedure (green rounded block).

in the MATLAB domain for each particle by replacing the old values with the new ones, whereas constant parameters (unflagged) are simply left unchanged; the file is then loaded in the NX domain. Since the evaluation of particles is independent, the description of the following steps up to the "vectorization" one will be focused on the processing of a single particle. Furthermore, it is worth mentioning that parallel processing of particles by means of multiple NX instances can be carried out, in order to improve the overall performance of the procedure.

(3) *Updating CAD template (NX).* The loaded parameters are applied to the features tree in order to generate the complete CAD model. The algorithm is also capable of identifying invalid CAD configurations (i.e. configurations that due to specific parameters values, result in an undesired topology or a NX update failure); in such cases, the corresponding solution particles are marked as negative results in the optimization routine by assigning a high objective function value to them and the subsequent steps can be avoided.

(4) *Computing CAD template – N STL objects distances (NX).* According to the matching list, the distances of the vertexes of each STL object from the associated CAD surfaces is evaluated. The processing is performed by using the *DeviationGauge* tool provided by the NX API (Siemens Documentation: Programming Tools, n.d) and generally represents the most expensive step of the entire procedure, also depending on the complexity of the *CAD template*. The NX *Deviation Gauge* tool is responsible for the identification of corresponding points; specifically, for each STL vertex, the software searches for the closest point on the corresponding CAD surface, within a given distance and an angular aperture w.r.t. the mesh normal direction. The sub-sampling of STL vertices in the distance evaluation step could be beneficial for the reduction of computing times and it is a strategy that will be tested in the future.

(5) *Passing distances from NX to MATLAB.* The distances computed in the previous step are exported by the *DeviationGauge* tool in *N* separated text files. Whenever a single evaluation fails (e.g. due to an invalid *CAD template* configuration, characterized by missing surfaces and an incorrect feature tree), an empty list of distances is generated in the related file; in this case, the entire particle is classified as invalid and the following steps can be skipped. The files are subsequently parsed by MATLAB, in order to import the distances in the MATLAB workspace.

(6) *Evaluating objective functions (MATLAB).* For each particle, the objective function is evaluated as in Eq. (1). In order to isolate degraded configurations, a dummy high value is ascribed to invalid particles.

(7) *Vectorization (MATLAB).* All the values of the evaluated objective functions are reported to a single vector, being each element related to a corresponding particle.

(8) *Updating best parameters set (MATLAB).* The minimum value of the vectorized objective functions is compared to the previous best result. If the former is lower, the corresponding particle is stored as the new best configuration, otherwise the latter is retained. At the first iteration, the best particle of the vectorized objective functions is simply stored.

(9) *Verifying final conditions.* The procedure is stopped if at least one of the exit condition is met in accordance to the particle swarm options provided; otherwise the flows proceeds to "updating swarm". Exit conditions are: (1) number of iterations have been exceeded a maximum value provided as an algorithm setting; (2) the value of objective function is below a provided threshold; (3) a stall condition is encountered, that is, the objective function's decrement over the last iterations is lower than a specified value.

(10) *Updating swarm.* The particles are randomly updated according to the PSO core algorithm and the provided options (the options influencing the most the PSO behavior in this specific application will be discussed in Section 3; a complete list of the algorithm parameters is available in Particle swarm optimization - MATLAB particleswarm - MathWorks Italia.)

3. Results and discussion

3.1. Overview

The TCRT has been implemented and tested on a computer equipped with: (i) Microsoft operating system (i.e. Windows 7), (ii) Siemens NX 10 and (iii) MATLAB R2016b. All the algorithms described in the previous section have been tested on a 128 GBs RAM workstation supplied with a six-core Intel® Xeon® E5-2643 v3 processor, which can manage up to 12 threads simultaneously at 3.40 GHz. All the tests hereby reported have been executed using 10 simultaneous NX instances running in parallel. Four CAD reconstruction tests starting from synthetic data (*Drilled Plate, Bracket, Pin, Flange*) and 1 from scanned data (i.e. *Electrical Socket Adapter*) have been performed (see Table 1 for details). Simple reconstruction tests have been included with propaedeutic purposes, hopefully to help the reader understanding the advantages of the described approach. The parts have been chosen and designed aiming at reproducing shapes and structures that are typical among mechanical parts. Both basic and advanced CAD features have been utilized in the modelling phase (e.g. extrusions, revolutions, circular arrays, etc.).

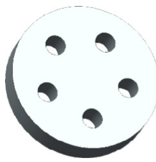
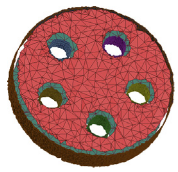
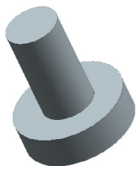
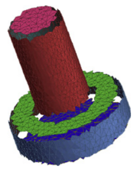
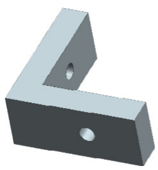
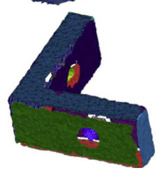
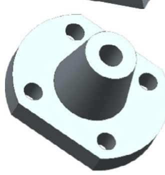
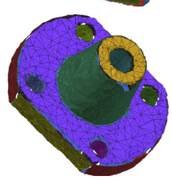
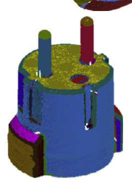
Specifically, the most basic topologies tested are defined by 4 parameters; for instance, the *Drilled Plate* is defined by *Extrusion Depth,* the *Base Diameter,* the *Hole Radial Distance* and the *Hole Diameter*, as depicted in Fig. 6. Instead, the largest number of parameters has been tested in the *Electrical Socket Adapter*, which is characterized by 20 parameters. For all tests, 7 additional parameters have been introduced to manage rigid transformations: rotations are controlled using 4 parameters in an axis-angle representation, whereas translations are described with X-Y-Z translations w.r.t. the coordinate system.

3.2. Input of the method: Reference data (mesh) and CAD model templates

The workflow for the creation of the synthetic reference data is shown in Fig. 7: a "perfect" STL model has been generated from the ideal CAD directly within Siemens NX 10, by setting both the triangle tolerance and the adjacency tolerance to 0.01 mm. The STL model has undergone a series of preparations steps simulating the data acquisition process: (i) a "remesh" step, in order to optimize the triangles distribution and dimensions; (ii) a decimation step; (iii) a corruption step, where the STL is impaired by two different error sources, i.e. a random zero-mean Gaussian noise, having standard deviation σ = 0.10 mm, that simulates the acquisition process and a human-introduced error that accounts for manufacturing defects; this last operation is manually performed in a mesh-editing software by perturbing the STL vertices positions at most by 0.3 mm (simulating the accuracy of a realistic 3D optical scanner); (iv) a segmentation step, which is interactively carried out by the user with the help of dedicated software tools.

In the specific case of the *Electrical Socket Adapter,* real scanned geometry has been obtained following a different process: the raw data has been acquired using a 3D optical scanner (i.e. a Romer RS1 mounted on a 7520-SI absolute arm by hexagon metrology

Table 1
CAD models tested with the TCRT: original CAD models and reference data.

Original CAD Model	Segmented mesh	Number of parameters	Mesh type	Number of triangles	Number of segmented regions
Drilled Plate 		11	Synthetic	12,964	8
Pin 		11	Synthetic	5210	5
Bracket 		13	Synthetic	4850	10
Flange 		16	Synthetic	6308	13
Electrical Socket Adapter  [a]		27	Scanned	147,532	40

[a] Actual physical model of the *Electrical Socket Adapter* that has been scanned to generate the reference data.

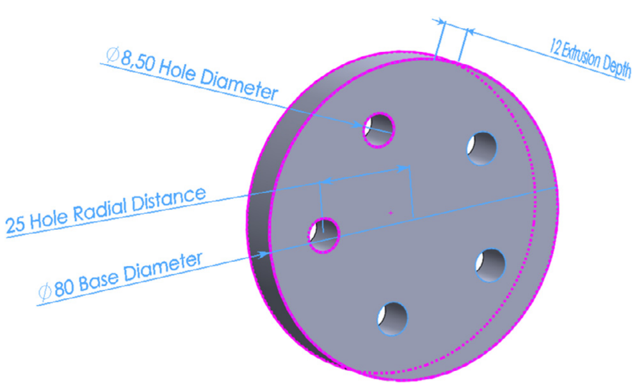


Fig. 6. *Drilled Plate* ideal CAD model: parameters and corresponding dimensions [mm].

(ROMER Absolute Arm with integrated scanner, n.d); the resulting mesh has been decimated keeping the number of triangles significantly higher than the synthetic STLs, in order to test the ability of TCRT to handle dense meshes.

In all cases a priori knowledge on the object shape and topology has been subsequently used to generate the *CAD templates* to be used in the fitting process by means of NX modelling tools; for tests on synthetic data, the feature tree used for the realization of the perfect STL model, easily deducible even from the impaired STL, has been adopted; conversely, the template of the *Electrical Socket Adapter* has been designed relying on both the observation of the scanned data and the expertise of the reverse engineer. Plausible dimensions completely defining the models, although not directly inferred from the reference data, were chosen and introduced. The template models used as starting point for the reconstruction are depicted in Fig. 8, along with the corresponding STL models. The positioning of showed CAD/STLs are the results of the preliminary registration phase; as can be appreciated in the figure, results were satisfactory in all the tested cases. In the following paragraphs, the results obtained applying the TCRT reconstruction on both synthetic and real-scanned data, respectively, are presented.

3.3. Reconstruction from synthetic data

The configuration of the PSO algorithm's settings (Particle swarm optimization - MATLAB particleswarm - MathWorks Italia, n.d) has been kept constant for all the tested case. Specifically, the *MinNeighborsFraction*[3] and *MaxStallIterations*[4] values were set to 0.75 e 20, respectively. The only exceptions are represented by the *SwarmSize* and the *InertiaRange*, which define the dimension and dynamicity of the swarm, respectively: such values have been adapted case-by-case.

For each test case, the specific optimization settings and the results obtained at the end of both the optimization stages (i.e. first

[3] *MinNeighborsFraction:* Minimum adaptive neighborhood size, a scalar from 0 to 1.
[4] *MaxStallIterations:* Iterations end when the relative change in best objective function value over the last *MaxStallIterations* iterations is less than an imposed tolerance value.

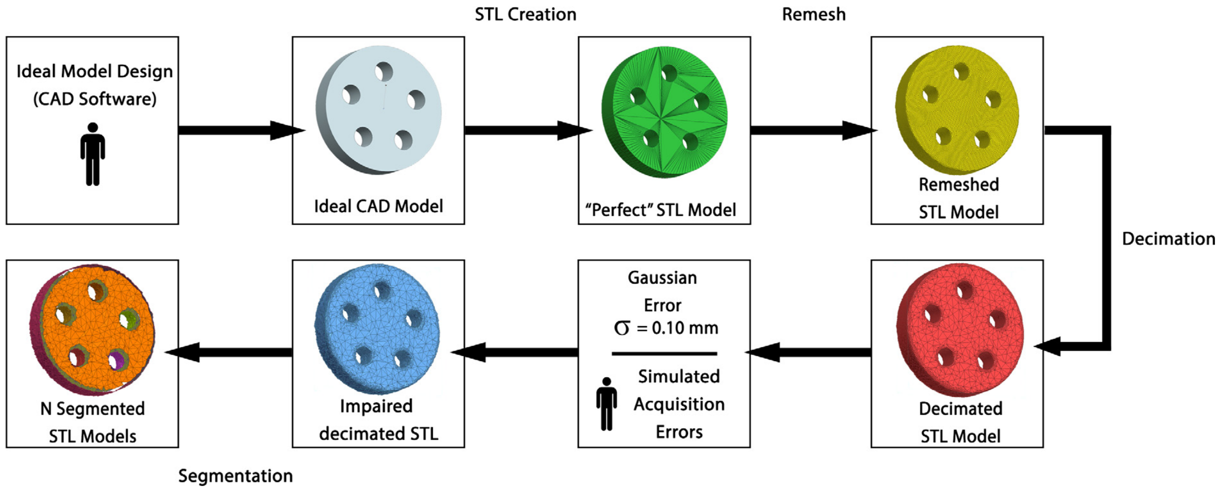


Fig. 7. Synthetic mesh creation process (*Drilled Plate* test case).

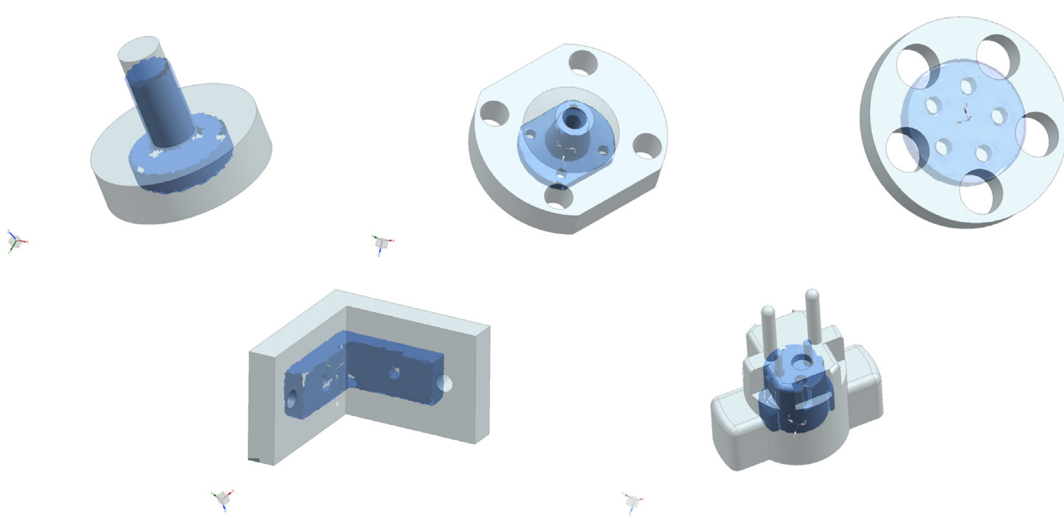


Fig. 8. Template CAD Models (grey) aligned with the corresponding reference STL models (blue) at the end of the preliminary phase described in Section 2.

and second optimization stages, which respectively make use of Eqs. (3) and (2) in Eq. (1)), are reported in Table 2. Incidentally, all optimization stages ended under the same termination criterion, i.e. value of the objective function has not improved over *MaxStallIterations* consecutive iterations. Moreover, the obtained results in terms of (i) execution time, (ii) iterations and (iii) final values of the objective functions are reported in Table 2 as well.

Some general considerations may be drawn from the analysis of the results: given enough computational power to the PSO algorithm, a CAD configuration sufficiently close to the global optimum (i.e. the original CAD model) has always been achieved. The optimization settings have been empirically determined, increasing the size of the particle swarm and tweaking the *InertiaRange* whenever needed. Generally, a higher value of *InertiaRange* has been considered and tested as beneficial with the increase of the number of parameters describing the CAD model topology. Unfortunately, even considering basic parametric representations, the time required to complete the reconstruction process using the developed tool is significant (the lowest value being 159 min achieved in the *flange* reconstruction). The second optimization phase has proved essential in all the tests, as it has always remarkably improved the result and drawn the reconstructed dimensions near to the corresponding original values.

The dimensions of the reconstructed CAD models have been compared with the original ones to evaluate the reconstruction errors; for instance, in Table 3, reconstructed and original values of the parameters defining the shape of the *flange* have been reported[5]. The reconstruction of the synthetic models have been also carried out independently by two experienced reverse engineers, in order to test the performance of the TCRT with respect to an interactive reconstruction approach based on state of the art software packages; specifically, the environment and tools offered by a renowned RE software (i.e. Rapidworks® RapidWorks: Reverse Engineering Software, n.d – a NextEngine version of Geomagic Design X®) have been used to perform the reconstruction. The reverse engineers have shared the same information on the reconstructed object that have been used to generate the *CAD templates* driving the TCRT reconstruction (e.g. symmetries, geometric constraints, manufacturing constraints, etc.) and, accordingly, they have tried to achieve correct feature trees and enforce correct geometric relations.

A comparison of the absolute and relative errors[6] measured in the four test cases following both the reconstruction approaches is

[5] Rotation and translation parameters were excluded as they were judged not interesting to assess the quality of the reconstruction.
[6] Both the errors are considered in magnitude (absolute values).

Table 2
Optimization settings and algorithm results for the reconstruction tests starting from synthetic data.

					First optimization phase			Second optimization phase		
CAD Model Name	Number of Parameters	Number of triangles	SwarmSize	InertiaRange	Time [min]	Function value after first optimization phase [mm]	Number of iterations	Time [min]	Function value after second optimization phase [mm]	Number of iterations
Drilled Plate	11	12,964	100	[0.2 1.1]	114	0.3839	101	73	0.0857	54
Pin	11	5210	100	[0.3 1.1]	435	0.7788	150	360	0.1151	117
Bracket	13	4850	200	[0.2 1.1]	634	1.415	212	651	0.1883	177
Flange	16	6308	200	[0.3 1.1]	87	1.6528	160	72	0.1865	91

Table 3
Dimensions of the reconstructed features in the *Flange* model: TCRT vs SoA reconstruction performed by two reverse engineers. The parameter "*Extrusion2_draftangle*" is an angle and corresponding values are to be intended in degrees.

Flange: original CAD		TCRT				Reverse engineer					
						#1			#2		
Parameter name	Correct value [mm]	Value after first optimization phase [mm]	Value after second optimization phase [mm]	Absolute error [mm]	Relative error	Reconstructed value [mm]	Absolute error [mm]	Relative error	Reconstructed value [mm]	Absolute error [mm]	Relative error
Arrayholes_distance	38.00	38.22	38.06	−0.06	0.001	38.25[a]	−0.25	0.007	38.02	−0.016	0.000
Base_diameter	100.00	98.83	99.79	0.21	0.002	99.90	0.10	0.001	99.56	0.443	0.004
Centralhole_diameter	15.00	14.94	14.95	0.05	0.003	14.79	0.21	0.014	14.88	0.122	0.008
Extrusion1	15.00	14.69	14.93	0.07	0.004	14.78	0.22	0.014	15.06	−0.058	0.004
Extrusion2	40.00	39.85	39.90	0.10	0.002	40.03	−0.03	0.001	40.51	−0.510	0.013
Extrusion2_draftangle	10.00	11.65	10.27	−0.27	0.027	10.15	−0.15	0.015	10.12	−0.116	0.012
Holes_diameter	11.00	11.24	10.97	0.03	0.002	10.92[a]	0.08	0.007	10.93	0.067	0.006
Side_flange	40.00	38.51	40.11	−0.11	0.003	39.95	0.05	0.001	39.83	0.172	0.004

[a] Values averaged across the corresponding CAD features, due to a non-perfect topology definition in the reconstructed model.

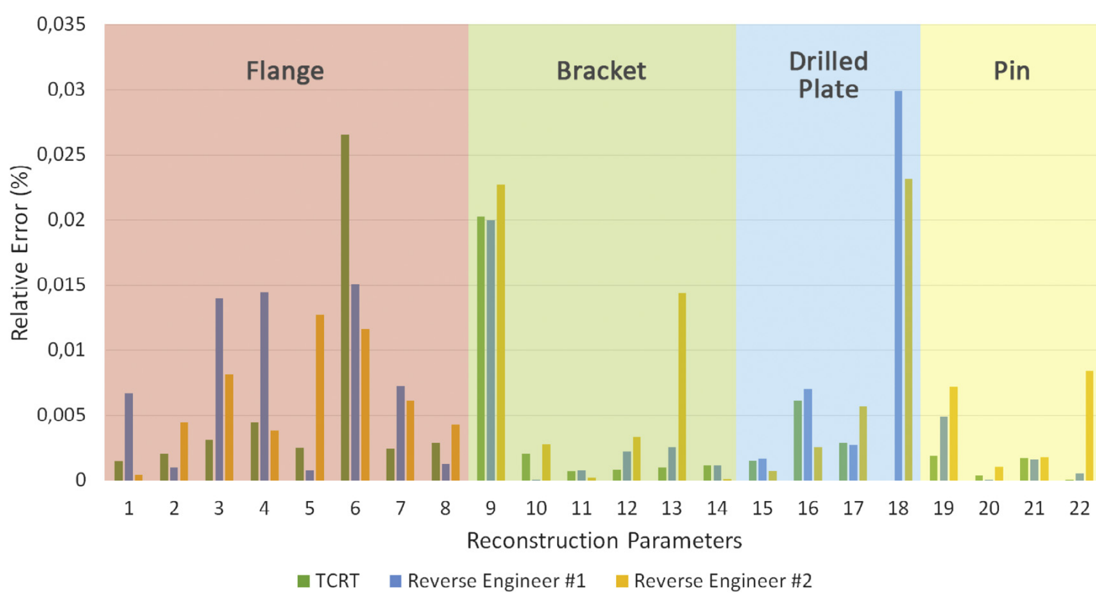


Fig. 9. Comparison of relative reconstruction errors obtained for *Pin, Flange, Drilled Plate, Bracket* models across each parameter: TCRT (green) vs traditional approach (Blue: Reverse Engineer #1, Yellow: Reverse Engineer #2).

reported in Fig. 9 and in Fig. 10. Noticeably, the TCRT performed comparably to the standard approach throughout all the tests, producing lower relative errors on the majority of parameters. Some statistics on the absolute and relative errors evaluated for each reconstructed model are reported in Table 4; moreover, the global statistics evaluated upon the 22 geometric parameters across all the models are summarized in the bottom row. Overall, the TCRT exhibits lower median and mean values considering both the absolute and the relative errors.

As extensively discussed in Section 1, the reconstructed topology (i.e. feature tree of the model, complete of all geometric constraints) also influences the usefulness of the final result. In this regard, the possibilities of the traditional approach have proved to be rather limited. Although being aware of the desirable result in terms of geometric constraint and relations to be enforced, designers usually must settle for a compromise solution, in order to use the tools provided by the RE software. A significant example, with this respect, may be found in the *Flange* case (Fig. 11): the

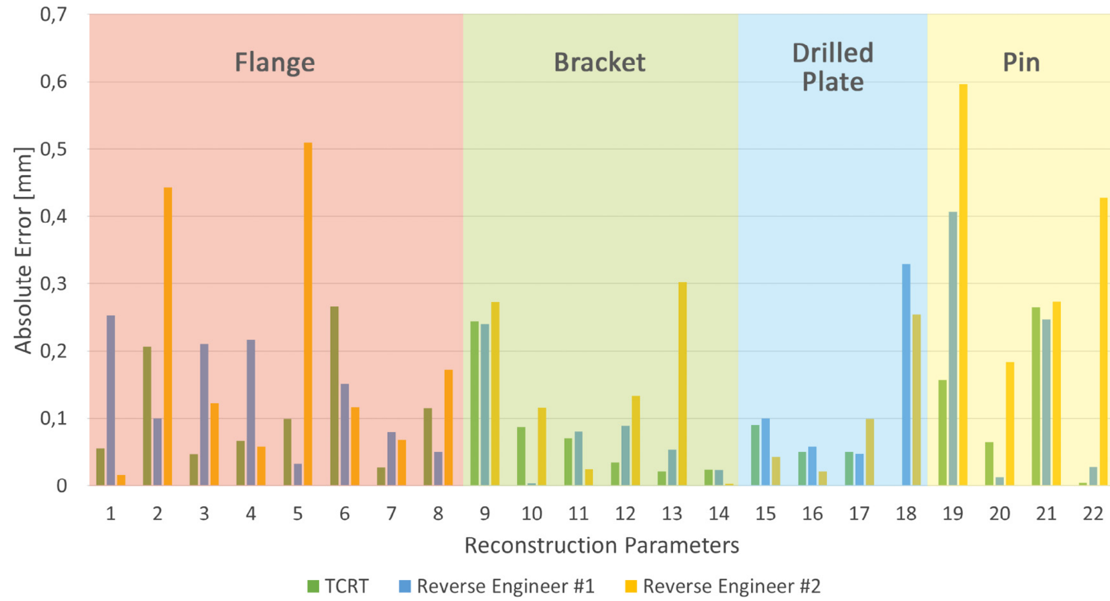


Fig. 10. Comparison of absolute reconstruction errors obtained for *Pin, Flange, Drilled Plate, Bracket* models across each parameter: TCRT (green) vs SoA approach (Blue: Reverse Engineer #1, Yellow: Reverse Engineer #2). Parameter #6 is an angle evaluated in degrees.

Table 4
Statistics of the absolute and relative reconstructions errors affecting *Flange, Pin, Bracket, Drilled Plate* models. Global statistics are reported in the last row. For the *Flange* model, the angular parameter has not been considered in the evaluation of the statistics for convenience.

		Absolute error [mm]					Relative error				
		Min	Max	Median	Mean	Dev. Std.	Min	Max	Median	Mean	Dev. Std.
Flange	TCRT	0.0272	0.2065	0.0664	0.0882	0.0603	0.0015	0.0044	0.0025	0.0027	0.0009
	Reverse Engineer #1	0.0319	0.2533	0.1000	0.1346	0.0898	0.0008	0.0145	0.0067	0.0065	0.0059
	Reverse Engineer #2	0.0163	0.4428	0.1224	0.1983	0.1973	0.0004	0.0128	0.0044	0.0057	0.0039
Pin	TCRT	0.0041	0.2646	0.1108	0.1226	0.1138	0.0001	0.0019	0.0011	0.0010	0.0009
	Reverse Engineer #1	0.0130	0.4064	0.1377	0.1737	0.1886	0.0001	0.0049	0.0011	0.0018	0.0022
	Reverse Engineer #2	0.1830	0.5968	0.3506	0.3702	0.1817	0.0010	0.0084	0.0045	0.0046	0.0037
Bracket	TCRT	0.0213	0.2433	0.0524	0.0799	0.0843	0.0007	0.0203	0.0011	0.0043	0.0078
	Reverse Engineer #1	0.0036	0.2395	0.0669	0.0815	0.0840	0.0001	0.0200	0.0017	0.0045	0.0076
	Reverse Engineer #2	0.0026	0.3020	0.1246	0.1418	0.1239	0.0001	0.0227	0.0030	0.0073	0.0092
Drilled Plate	TCRT	0.0000	0.0900	0.0500	0.0475	0.0369	0.0000	0.0061	0.0022	0.0026	0.0026
	Reverse Engineer #1	0.0476	0.3290	0.0791	0.1337	0.1322	0.0017	0.0299	0.0049	0.0103	0.0133
	Reverse Engineer #2	0.0210	0.2546	0.0714	0.1046	0.1054	0.0007	0.0231	0.0041	0.0080	0.0103
Global	TCRT	0.0000	0.2647	0.0641	0.0846	0.0748	0.0000	0.0203	0.0019	0.0028	0.0043
	Reverse Engineer #1	0.0036	0.4064	0.0801	0.1267	0.1150	0.0001	0.0299	0.0022	0.0058	0.0078
	Reverse Engineer #2	0.0026	0.5968	0.1337	0.1971	0.1754	0.0001	0.0232	0.0043	0.0064	0.0067

reverse engineer #1 has deliberately chosen to neglect symmetries in the holes array, performing an independent fitting of each hole in order to try to minimize the errors in the identification of the holes diameters. Nevertheless, the reconstructed diameters (see Table 3) are comparable on average with the values obtained by reverse engineer #2; moreover, the final model is affected by an incorrect topology (i.e. the holes positions are not symmetric w.r.t. the central axis – see Fig. 11b). The TCRT reconstruction, instead, performs slightly worse in the identification of the holes dimensions, but globally outperforms the other ones (see Mean and Dev.Std. columns in Table 4) and is characterized by a correct imposition of symmetries and geometric relations (Fig. 11c). In fact, one of the great benefits of the proposed approach enabling to reach the described results, is that rigid transforms of the *CAD template* are allowed at each step of the fitting, seeking the best alignment to minimize global reconstruction errors. On the contrary, in the traditional RE software-based reconstruction, only a single alignment is initially performed exploiting symmetries (at best) and the main geometric features.

A deviation analysis (Fig. 12) has been additionally performed to analyse the error distribution on the reconstructed models relatively to the reference data; an allowable tolerance of ±0.05 mm, that is mapped in green in Fig. 12, has been set. Evidently, most significant errors may be identified across the edges of the models, which are the areas mostly affected by the STL creation process described in Section 3.2: in the *remesh* step the original sharp edges have been, in fact, heavily modified and blunted in the process to simulate a typical acquisition effect.

3.4. Reconstruction from real-scanned data

The *Electrical Socket Adapter* model has been reconstructed starting from data acquired on a real, physical object. Two main aspects distinguish this test from the ones previously described:

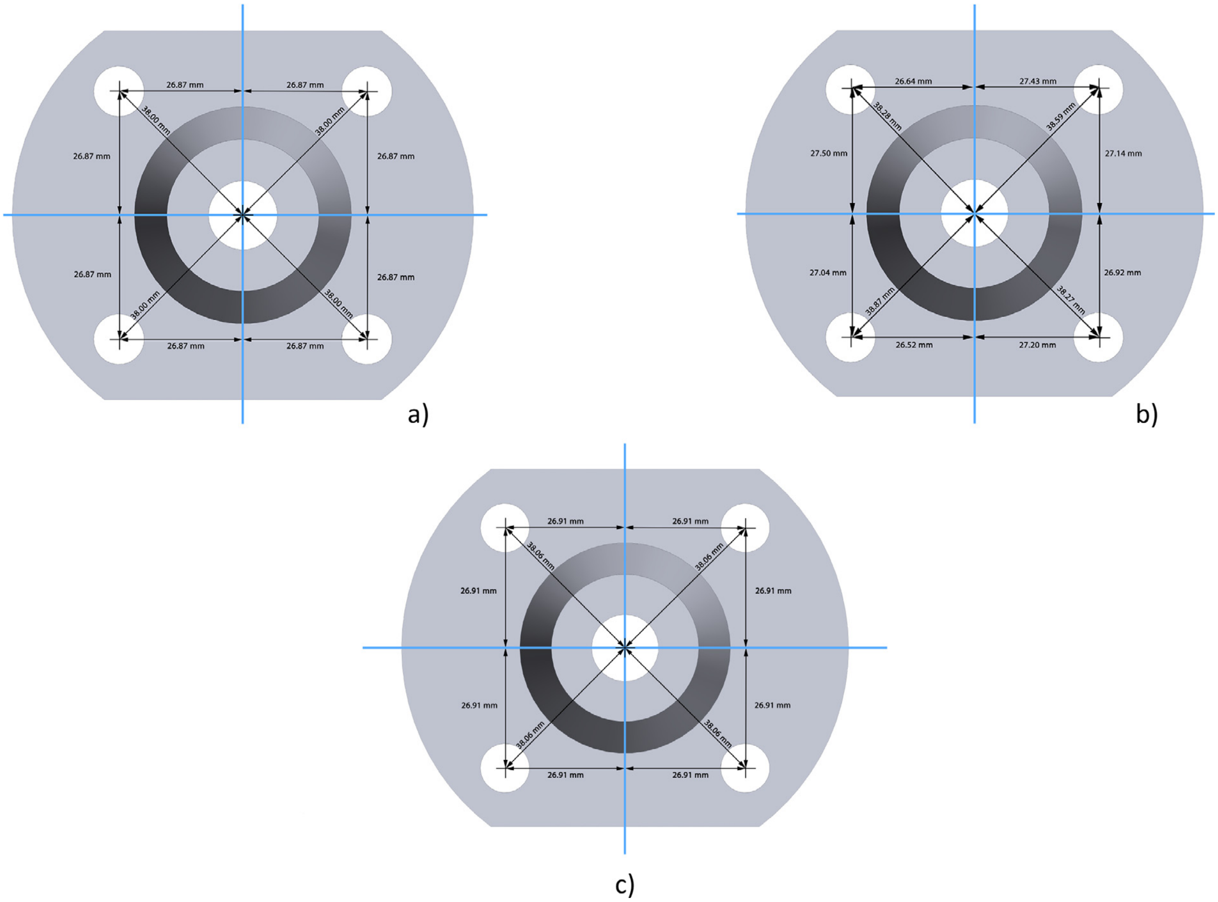


Fig. 11. Front view of the *Flange* model with dimensions of the 4-holes array. (a) original CAD model; (b) traditional reconstruction, performed within *Rapidworks*® by reverse engineer #1, affected by an incorrect topology; (c) reconstruction performed with the TCRT.

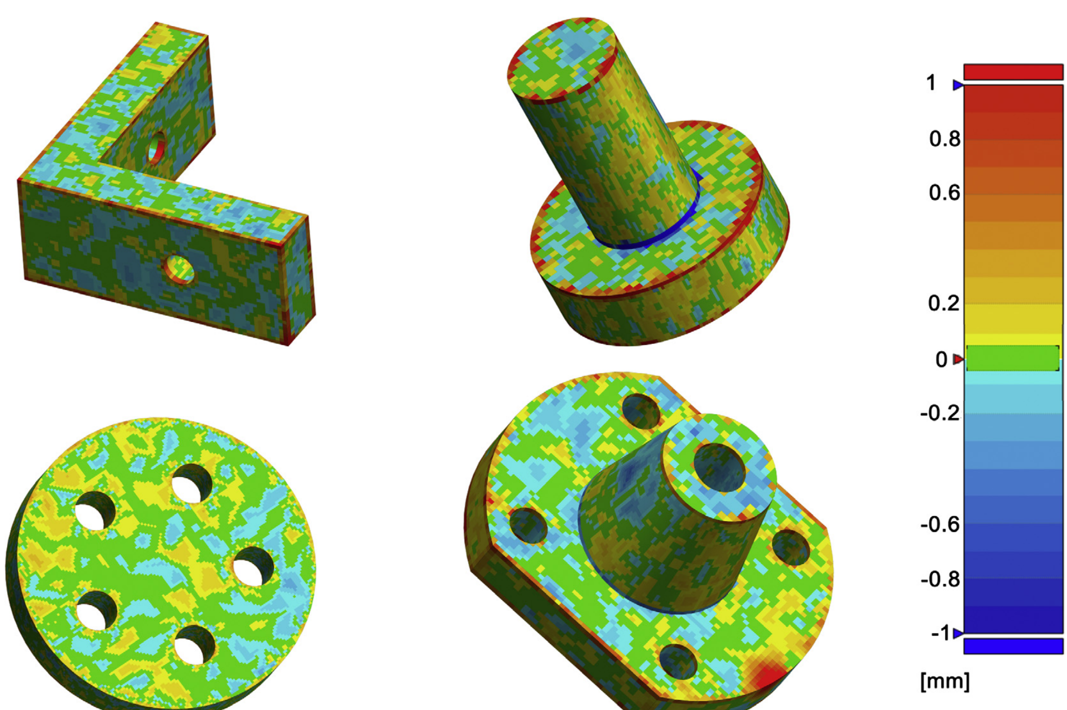


Fig. 12. Deviation maps of the four TCRT reconstructed models (from left to right, top to bottom): *Bracket*, *Pin*, *Drilled Plate* and *Flange*.

(1) a high density of triangles, which implies an overall greater number of triangles; (2) a significantly more complicated feature tree, completely defined by 27 parameters (details are reported in Table 1). Both these aspects have deeply influenced the algorithm, imposing additional difficulties.

The great number of points have heavily weighed down the second optimization phase (i.e. driven by Eq. (2)), resulting in a huge time per iteration needed (e.g. around 2 h per iteration, using a 350 particles swarm); the high computational costs are associated with the evaluation of the mean deviation value, which requires a complete parsing of the distance text files. Conversely, the evaluation of the max distance that is carried out according to Eq. (3) in the first optimization phase, is cheaper since it is directly provided by NX's *DeviationGauge* tool.

As a result, the optimization on the *Electrical Socket Adapter* has been carried out applying Eq. (3) in both the first and second stage; details on the settings used in both the optimization stages are reported in Table 5. Although Eq. (3) does not represent the ideal objective function, the obtained results are somehow satisfying: the algorithm correctly explored the high-dimensional solution space describing the shape of the model, providing a solution in the neighbourhood of the global solution. The final result, achieved at the end of both optimization phases, is reported in Fig. 13 aligned with the raw data.

The final CAD model shows macroscopic flaws which may be traced back to the incomplete exploration of the solution space, probably due to the high dimensionality induced by the 27 parameters defining the model. The most evident defects are reported in Fig. 14 and may be summarized in: (i) the overall height of the socket is sensibly greater in the reconstructed model; (ii) the angle of the surface connecting the side elements and the main cylinder is different. A deviation map, representing the error distribution obtained at the end of the reconstruction of the *Electrical Socket Adapter* is depicted in Fig. 15a.

Table 5
Optimization settings and algorithm results for the *Electrical Socket Adapter*.

Electrical Socket Adapter	Number of parameters Number of triangles	27 147,532
First Optimization Phase	SwarmSize InertiaRange Time Function Value after First Optimization Phase Number of Iterations	350 [0.2 1.1] 1434 min 15.021 mm 358
Second Optimization Phase	SwarmSize InertiaRange Time Function Value after Second Optimization Phase Number of Iterations	500 [0.2 1.1] 2760 min 0.9424 mm 192

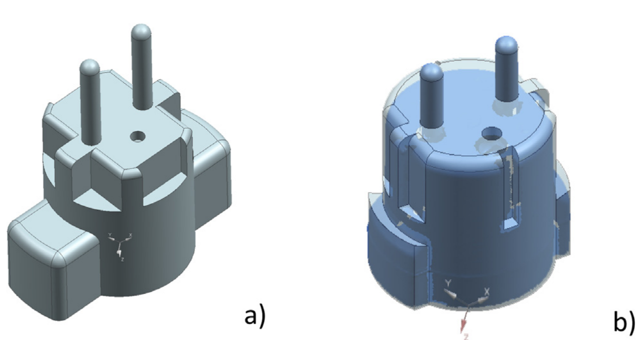


Fig. 13. *Electrical Socket Adapter* test case: (a) Initial CAD template (grey); (b) TCRT reconstruction result (grey), aligned with the acquired reference data (blue).

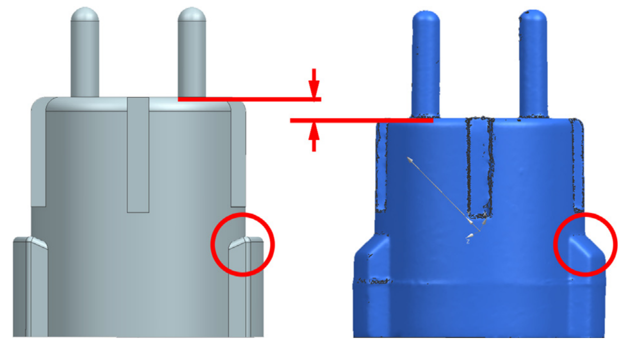


Fig. 14. Principal reconstruction errors for the *Electrical Socket Adapter* (highlighted in red): (1) erroneous position of the upper planar surface; (2) incorrect angle of the side extrusions.

An increase of the swarm size and the use of a higher *StallIterations* value may represent strategies to reach a better result by increasing the computational resources granted to the algorithm; however, it has to be noted that the inability of using Eq. (2) in this case represented the most limiting factor. The use of Eq. (3) in the final optimization phase is, in fact, not ideal: plausible errors in the acquisition and segmentation of data heavily affect the results (e.g. a single triangle erroneously segmented may jeopardize the correct retrieval of a model parameter). Possible solutions, with this respect, are represented by: (i) the study and application of a different objective function, capable to tolerate a higher number of triangles compatibly with the tools offered by NX and offering, at the same time, a more appropriate "guide" to the final result; (ii) a more significant decimation step, which may allow the use of Eq. (2) as in the tests previously discussed.

In order to prove the effectiveness of the latter strategy, a severe decimation step has been applied to the original reference data and a new reconstruction using the TCRT has been performed, this time using Eq. (2) as metric within the Eq. (1). The retrieved model is depicted in Fig. 15b; as it can be noted, the reconstruction result is significantly improved considering the dimensional accuracy; evidently, the optimization has been concluded without being trapped in the local minimum previously found.

4. Conclusions and future work

In this paper a novel method for the reconstruction of CAD models starting from pre-segmented mesh data has been presented; the key-factor of the proposed framework is the generation of a fully editable parametric CAD model, dimensionally faithful to the reference data and perfectly compliant to any type of geometrical constraint.

The proposed implementation partially relies on a user-guided framework to exploit *a priori* known information about the object to be reconstructed as well as the expertise of the designer in charge of the reconstruction process. A template model, parametrically designed using a CAD software package, is used as input to provide a complete description of the object topological information. As a matter of fact, there are theoretically no restriction on the number and type of CAD parameters the reverse engineer can adopt.

The proposed method has been shown to perform slightly better with respect to SoA commercial RE software tools in terms of dimensional accuracy of the reconstructed objects. Moreover, compared with the traditional *constrained fitting* approaches, it allows the retrieval of "perfect" CAD models, i.e. with *hard constraints* flawlessly imposed. This aspect is a long-term goal of RE approaches (see *Beautification* approaches (Gao et al., 2004;

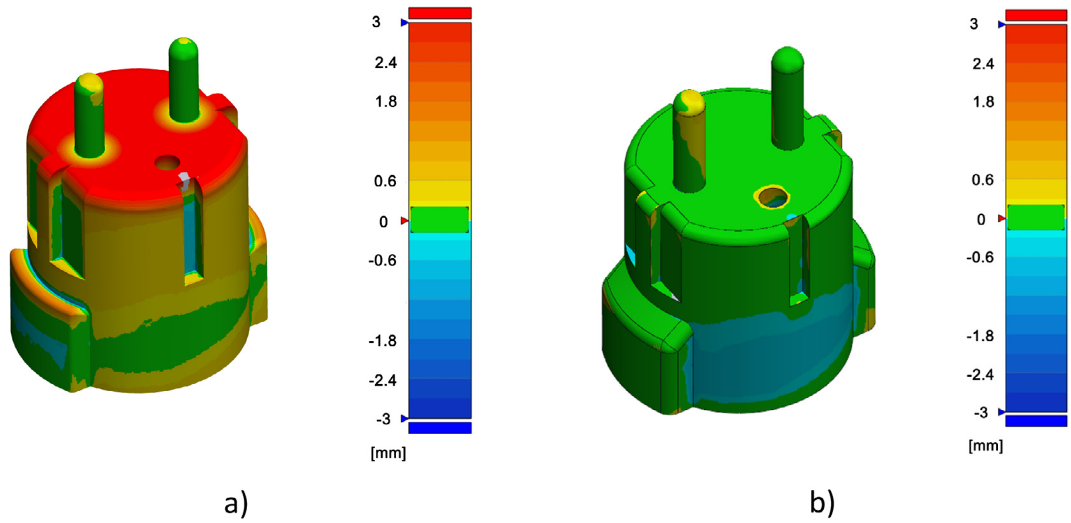


Fig. 15. Deviation map of the *Electrical Socket Adapter* test case reconstructed using the TCRT. (a) original mesh (b) decimated *mesh.*

Langbein et al., 2001; Langbein et al., 2004) and it represents one of the main achievements of the described framework. Moreover, the retrieval of an associative-parametric modelling history is an achievement that entails multiple advantages in CAD applications (Wang, Gu, Yu, Tan, & Zhou, 2012).

From the point of view of the practical applicability, we have imagined two possible usage scenarios. In a first situation, relying on the traditional software-based RE framework, the described method could be integrated as an additional fitting step to be performed at the end of the reconstruction. The parametric CAD model generated by means of RE software could be used as template in the TCRT. This "refinement" step could minimize the overall global reconstruction error without imposing potentially any additional user involvement; moreover, the CAD template would be already closely fitted to the reference data, hence heavily abating the computational costs of the optimization phase and likely reducing the convergence to spurious local minima.

In an alternative situation, the template could come from an a priori available collection of CAD models. In such a case, the template dimensions and proportions could significantly differ from the ones of the mesh data with a consequent exacerbation of the issues related to computation time and convergence solution.

In both the described scenarios the presented method could be applied with significant advantages in terms of "quality" of the final result (i.e. a CAD model that is in compliance with known geometric constraints and dimensionally accurate) and manual work required to the reverse engineer. Generally, it must be noted that the effectiveness of the proposed method relies on the available knowledge of the part to be reconstructed. Major limitations may be perceived in the need for the user to provide a custom CAD template for each object to be reconstructed. However, regardless of the chosen application scenario, the proposed framework empowers the user with a flexible tool: a first attempt template can be initially used; the obtained result can be eventually adjusted (i.e. editing the feature tree, adding or removing geometric constraints, changing the parameters included in the optimization routine) and used as starting point for a new optimization.

As far as the implementation of the described method is concerned, there is room for improvements referring both to the preliminary and the optimization phases. Indeed, the former significantly relies on user interaction, especially for the template design and the matching between the CAD and the acquired data. It is authors' opinion that the reverse engineer's contribution is precious and irreplaceable for the generation of the template and the determination of its topology; however, different approaches might be hypothesized to reduce the user's burden. For instance, semantic mapping of 3D data (Günther, Wiemann, Albrecht, & Hertzberg, 2015) might allow the recognition of a CAD template corresponding to a scanned object directly from a pre-built library of CAD models, implementing, in fact, a CAD model retrieval (Gao, Li, & Zhang, 2015; Paterson, Corney, Sherlock, & Mill, 2015; Zehtaban, Elazhary, & Roller, 2016) process. Furthermore, given a valid segmentation of the 3D original data, graph isomorphism techniques are deemed to be able to retrieve the corresponding CAD object from a model database and, analysing connected surfaces and their geometric properties, even to perform the matching between corresponding surfaces (Ip & Gupta, 2007). An interesting development could be achieved by introducing a CAD template with a fully editable modelling history and allowing the optimization routine to introduce or remove additional geometric features; a promising implementation of such strategy would be to rely on a master model as suggested in Zheng, Zhang, Gao, and Wang (2016), realizing a variational design approach.

Considering the optimization phase, the adoption of an unconventional error metric (i.e. Eq. (3)) rather than the more common mean Euclidean distance (i.e. Eq. (2)) and the choice of the PSO algorithm have been proved to be effective for the considered application. It is worthy to note that the proposed approach does not rely on a particular class of optimization algorithms; thus, future tests will be oriented towards the implementations of alternative population-based procedures (Dorigo, Di, & Gambardella, 1999; Karaboga & Basturk, 2007) (which, in our opinion, represent the class of algorithms best suited to tackle the described problem) or the development of ad-hoc strategies and metrics.

The efficiency of the proposed method is a critical factor for its applicability. The current computational time needed on the tested workstation is significant higher if compared to commercial reconstruction tools. Efficiency improvements might be achieved by conveniently tuning both the optimization algorithm and the distance metric. Nevertheless, more than 90% of the current computational cost is ascribable to two external sources: (i) the internal NX processing, needed to compute the required distances, (ii) passing the distances' values from NX to MATLAB. It is authors' hope that these limitations might be amended by relying on NX future releases (allowing, for instance, to skip the writing/reading of a text file to pass parameters from NX to MATLAB and vice versa) or by using alternative commercial software packages for the distances computation step.

Finally, future tests will be oriented towards the reconstruction of models composed also by freeform surfaces that usually introduce additional computational complexity due to the higher number of controlling parameters; with this respect, satisfying results have been obtained by using genetic algorithms (e.g. 2D spline fitting in Yoshimoto, Harada, and Yoshimoto (2003)) or PSO itself (Gálvez & Iglesias, 2012), proving the effectiveness of nature-inspired algorithms when dealing with data fitting problems in RE.

Conflicts of interest

There is no conflict of interest related to this work.

References

3D Systems (2016). Geomagic Design X (formerly Rapidform XOR). <http://www.rapidform.com/products/xor/overview/> Accessed 29.09.16.

Anwer, N., & Mathieu, L. (2016). From reverse engineering to shape engineering in mechanical design. *CIRP Annals - Manufacturing Technology, 65*, 165–168. https://doi.org/10.1016/j.cirp.2016.04.052.

Au, C. K., & Yuen, M. M. F. (1999). Feature-based reverse engineering of mannequin for garment design. *Computer-Aided Design, 31*, 751–759. https://doi.org/10.1016/S0010-4485(99)00068-8.

Benko, P., Kós, G., Várady, T., Andor, L., & Martin, R. (2002). Constrained fitting in reverse engineering. *Computer Aided Geometric Design, 19*, 173–205. https://doi.org/10.1016/S0167-8396(01)00085-1.

Bey, A., Chaine, R., Marc, R., Thibault, G., & Akkouche, S. (2012). Reconstruction of consistent 3D CAD models from point cloud data using a priori CAD models. In *ISPRS – International Archives of the Photogrammetry, Remote Sensing and spatial information sciences* (pp. 289–294). https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-289-2011. XXXVIII-5.

Bosché, F. (2010). Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics, 24*, 107–118. https://doi.org/10.1016/j.aei.2009.08.006.

Buonamici, F., & Carfagni, M. (2016). 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, ASME*. https://doi.org/10.1115/DETC2016-59242, p. V01BT02A003.

Burston, M., Sabatini, R., Gardi, A., & Clothier, R. (2014). Reverse engineering of a fixed wing Unmanned Aircraft 6-DoF model based on laser scanner measurements. In *2014 IEEE Metrology for Aerospace (MetroAeroSpace), IEEE* (pp. 144–149). https://doi.org/10.1109/MetroAeroSpace.2014.6865910.

Chang, K. H., & Chen, C. (2011). 3D shape engineering and design parameterization. *Computer-Aided Design and Applications, 8*, 681–692. https://doi.org/10.3722/cadaps.2011.681-692.

Dorigo, M., Di, Caro G., & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life, 5*, 137–172. https://doi.org/10.1162/106454699568728.

Durupt, A., Remy, S., Ducellier, G., & Pouille, P. (2014). Reverse engineering using a knowledge-based approach. *International Journal of Product Development, 19*, 113. https://doi.org/10.1504/IJPD.2014.060045.

Erdős, G., Nakano, T., & Váncza, J. (2014). Adapting CAD models of complex engineering objects to measured point cloud data. *CIRP Annals – Manufacturing Technology, 63*, 157–160. https://doi.org/10.1016/j.cirp.2014.03.090.

Fayolle, P. A., & Pasko, A. (2015). User-assisted reverse modeling with evolutionary algorithms. In *2015 IEEE congress on evolutionary computation, CEC 2015 – Proceedings 2015* (pp. 2176–2183). https://doi.org/10.1109/CEC.2015.7257153.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*, 381–395. https://doi.org/10.1145/358669.358692.

Fisher, R. B. (2002). Applying knowledge to reverse engineering problems. *Proceedings - Geometric Modeling and Processing: Theory and Applications, GMP, 2002*(36), 149–155. https://doi.org/10.1109/GMAP.2002.1027506.

Gálvez, A., & Iglesias, A. (2011). Efficient particle swarm optimization approach for data fitting with free knot -splines. *Computer-Aided Design, 43*, 1683–1692. https://doi.org/10.1016/j.cad.2011.07.010.

Gálvez, A., & Iglesias, A. (2012). Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Information Sciences, 192*, 174–192. https://doi.org/10.1016/j.ins.2010.11.007.

Gao, C. C. H., Langbein, F. F. C., Marshall, A. D. A., & Martin, R. R. R. (2004). Local topological beautification of reverse engineered models. *Computer-Aided Design, 36*, 1337–1355. https://doi.org/10.1016/j.cad.2004.02.004.

Gao, Xueyao, Li, Huinan, & Zhang, Chunxiang (2015). A CAD model retrieval tool based on face attribute adjacent graph. In *2015 4th international conference on computer science and network technology (ICCSNT)* (pp. 117–120). IEEE. https://doi.org/10.1109/ICCSNT.2015.7490718.

Geng, Z., & Bidanda, B. (2017). Review of reverse engineering systems – current state of the art. *Virtual and Physical Prototyping, 12*, 161–172. https://doi.org/10.1080/17452759.2017.1302787.

Günther, M., Wiemann, T., Albrecht, S., & Hertzberg, J. (2015). Model-based furniture recognition for building semantic object maps. *Artificial Intelligence, 1*, 1–16. https://doi.org/10.1016/j.artint.2014.12.007.

Holland, J. H. (1995). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Hosseini, S. F., & Moetakef-Imani, B. (2017). Innovative approach to computer-aided design of horizontal axis wind turbine blades. *Journal of Computational Design and Engineering, 4*, 98–105. https://doi.org/10.1016/j.jcde.2016.11.001.

Ip, C. Y., & Gupta, S. K. (2007). Retrieving matching CAD models by using partial 3D point clouds. *Computer-Aided Design and Applications, 4*, 629–638. https://doi.org/10.1080/16864360.2007.10738497.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471. https://doi.org/10.1007/s10898-007-9149-x.

Kennedy, J., Eberhart, R. (n.d.). Particle swarm optimization. In *Proceedings of ICNN'95 – international conference on neural networks* Vol. 4 (p. 1942–8). IEEE. https://doi.org/10.1109/ICNN.1995.488968.

Langbein, F. C., Marshall, A. D., & Martin, R. R. (2004). Choosing consistent constraints for beautification of reverse engineered geometric models. *Computer-Aided Design, 36*, 261–278. https://doi.org/10.1016/S0010-4485(03)00108-8.

Langbein, F. C., Mills, B. I., Marshall, A. D., & Martin, R. R. (2001). Recognizing geometric patterns for beautification of reconstructed solid models. In *Proceedings – international conference on shape modeling and applications* (pp. 10–19). SMI. https://doi.org/10.1109/SMA.2001.923370.

Locatelli, M., & Schoen, F. (2013). *Global optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611972672.

Michalewicz, Z. (1997). *Genetic algorithms + data structures = evolution programs* (vol. 24). Springer-Verlag. https://doi.org/10.1016/S0167-9473(97)87028-4.

Ohbuchi, R., Otagiri, T., Ibato, M., & Takei, T. (2002). Shape-similarity search of three-dimensional models using parameterized statistics. In *Proceedings – pacific conference on computer graphics and applications 2002* (pp. 265–274). https://doi.org/10.1109/PCCGA.2002.1167870.

Particle swarm optimization - MATLAB particleswarm - MathWorks Italia (n.d.) <https://it.mathworks.com/help/gads/particleswarm.html> Accessed 24.02.17.

Paterson, D., Corney, J., Sherlock, A., & Mill, F. (2015). Feature based search of 3D databases. In *ASME 2016 international design engineering technical conferences and computers and information in engineering conference*. https://doi.org/10.1115/DETC2016-60521, p. V01BT02A010.

RapidWorks: Reverse Engineering Software (n.d.) <http://www.nextengine.com/products/rapidworks/specs/overview> Accessed 03.02.17.

Robertson, C., Fisher, R., Werghi, N., & Ashbrook, A. P. (2000). Fitting of constrained feature models to poor 3d data. *Evolutionary Design and Manufacture*, 149–160. https://doi.org/10.1007/978-1-4471-0519-0_12.

ROMER Absolute Arm with integrated scanner (n.d). <http://www.hexagonmi.com/products/portable-measuring-arms/romer-absolute-arm-with-integrated-scanner> Accessed 25.01.17.

Siemens Documentation: Programming Tools (n.d.) <https://docs.plm.automation.siemens.com/tdoc/nx/10/nx_api/#uid:index> Accessed 24.02.17.

Solaberrieta, E., Minguez, R., Barrenetxea, L., Sierra, E., & Etxaniz, O. (2014). Computer-aided dental prostheses construction using reverse engineering. *Computer Methods in Biomechanics and Biomedical Engineering, 17*, 1335–1346. https://doi.org/10.1080/10255842.2012.745859.

Várady, T., Martin, R. R., & Cox, J. (1997). Reverse engineering of geometric models—an introduction. *Computer-Aided Design, 29*, 255–268. https://doi.org/10.1016/S0010-4485(96)00054-1.

Voicu, A. C., Gheorghe, I. G., Badita, L. L., & Cirstoiu, A. (2013). 3D Measuring of complex automotive parts by multiple laser scanning. *Applied Mechanics and Materials, 371*, 519–523. https://doi.org/10.4028/www.scientific.net/AMM.371.519.

Wang, J., Gu, D., Gao, Z., Yu, Z., Tan, C., & Zhou, L. (2013). Feature-based solid model reconstruction. *Journal of Computing and Information Science in Engineering, 13*, 11004. https://doi.org/10.1115/1.4023129.

Wang, J., Gu, D., Yu, Z., Tan, C., & Zhou, L. (2012). A framework for 3D model reconstruction in reverse engineering. *Computers & Industrial Engineering, 63*, 1189–1200. https://doi.org/10.1016/j.cie.2012.07.009.

Werghi, N., Fisher, R., Robertson, C., & Ashbrook, A. (1999). Object reconstruction by incorporating geometric constraints in reverse engineering. *CAD Computer Aided Design, 31*, 363–399. https://doi.org/10.1016/S0010-4485(99)00038-X.

Yoshimoto, F., Harada, T., & Yoshimoto, Y. (2003). Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design, 35*, 751–760. https://doi.org/10.1016/S0010-4485(03)00006-X.

Zehtaban, L., Elazhary, O., & Roller, D. (2016). A framework for similarity recognition of CAD models. *Journal of Computational Design and Engineering, 3*, 274–285. https://doi.org/10.1016/j.jcde.2016.04.002.

Zheng, Q., Zhang, S., Gao, S., & Wang, C. (2016). Fast variational design based on a master model generated from part model series. In *Volume 1B: 36th computers and information in engineering conference*. https://doi.org/10.1115/DETC2016-59239, p. V01BT02A002.