Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/22962

To cite this version :

Ali SHAH GHAZANFAR, Franca GIANNINI, Marina MONTI, Jean-Philippe PERNOT, Arnaud POLETTE - User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models - Journal of Computing and Information Science in Engineering p.13 - 2021

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Arts et Métiers Institute of Technology

User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models

Ghazanfar Ali Shah*
Doctoral Student
LISPEN, EA 7515, HeSam
Arts et Métiers
Aix-en-Provence, 13617
France
Email: ghazanfar_920@yahoo.com

Arnaud Polette
Associate professor
LISPEN, EA 7515, HeSam
Arts et Métiers
Aix-en-Provence, 13617
France
Email: arnaud.polette@ensam.eu

Jean-Philippe Pernot†
Professor
LISPEN, EA 7515, HeSam
Arts et Métiers
Aix-en-Provence, 13617
France
Email: jean-philippe.pernot@ensam.eu

Franca Giannini
Research director
IMATI
Consiglio Nazionale delle Ricerche
Genova,16149
Italy
Email: franca.giannini@ge.imati.cnr.it

Marina Monti
Researcher
IMATI
Consiglio Nazionale delle Ricerche
Genova,16149
Italy
Email: marina.monti@ge.imati.cnr.it

This paper introduces a novel reverse engineering (RE) technique for the reconstruction of editable CAD models of mechanical parts' assemblies. The input is a point cloud of a mechanical parts' assembly that has been acquired as a whole, i.e. without disassembling it prior to its digitization. The proposed framework allows for the reconstruction of the parametric CAD assembly model through a multi-step reconstruction and fitting approach. It is modular and it supports various exploitation scenarios depending on the available data and starting point. It also handles incomplete datasets. The reconstruction process starts from roughly sketched and parameterized CAD geometries (i.e. 2D sketches, 3D parts or assemblies) that are then used as input of a simulated annealing-based fitting algorithm, which minimizes the deviation between the point cloud and the adapted geometries. The coherence of the CAD models is maintained by a CAD modeler that performs the geometries' updates while guaranteeing the possibly imposed constraints and model coherence. The optimization process leverages a two-level filtering technique able to capture and manage the boundaries of the geometries inside the overall point cloud in order to allow local fitting and interfaces detection. It is a user-driven approach where the user decides what are the most suitable steps and sequence to operate. It has been tested and validated on both real scanned point clouds and as-scanned virtually generated point clouds incorporating several artifacts that would appear with real acquisition devices.

1 Introduction

The availability of 3D digital models of real-world products, systems, buildings, environments or human bodies turns out to be of major interest for the development of new applications and services. This has been possible due to the rapid evolution of digitization technologies, and the interest has accelerated for the digital twin creation in the scope of Industry 4.0 [1]. For many years, the reconstruction and treatment of 3D models from point clouds has focused the attention of the research community, and notably

*Address all correspondence related to format and figures to this author.
†Address all correspondence for other issues to this author.

Copyright © by ASME

for what concerns 3D meshes [2]. Indeed, less attention has been paid to the generation or update of parametric CAD models, which are nevertheless considered as reference models used all along the Product Design Process (PDP). Today, existing reverse engineering approaches mostly consider time-consuming patch-by-patch reconstruction strategies. The resulting manifold B-Rep models are not always editable later and are to be considered as dead models that do not rely on real parametric building trees [3]. Engineering knowledge about the shape of the object that needs to be reconstructed that could allow a more effective and accurate reconstruction [4] is not exploited during the RE process (e.g. geometric regularities, known dimensions, etc.). Inadequate attention is also paid to the reverse engineering of mechanical parts' assemblies that have been acquired as a whole, i.e. without disassembling them prior to their digitization. However, this is one of the prerequisites to enable the creation of digital twins capable of tracking physical systems for advanced industrial applications [5]. For instance, such a possibility would allow the update of the digital mock-up of a robot or production line to follow real-time changes, to keep the coherence between the virtual and physical worlds, and to support decision-making. Moreover, existing approaches do not allow for simultaneous multi-dimensional constraints satisfaction [6] (e.g. on 2D sketches, 3D features/parts, assemblies) and they often require almost complete point clouds.

This paper introduces a novel reverse engineering technique for the reconstruction of editable CAD models of mechanical parts' assemblies. From a point cloud of a mechanical parts' assembly acquired as a whole. The proposed framework allows reconstruction of the CAD assembly model through a multi-step approach based on a generic and multi-dimensional fitting technique. Depending on both the reverse engineering process start point and the available data, several exploitation scenarios are supported. The point cloud does not need to be complete as the approach does not fit the patches one-by-one. The reconstruction process starts from roughly sketched and parameterized geometries (i.e 2D sketches, 3D parts or assemblies) that are then adapted to fit the point cloud using a simulated annealing-based algorithm, that minimizes the deviation between the point cloud and the geometries. The segmentation of the point cloud and the proper identification of the interfaces between the geometries leverage a two-level filtering technique based on distance and normal computation. The coherence of the CAD models, whose parameters are modified at each optimization step, is maintained by a CAD modeler that performs the updates and satisfies the geometric constraints as the fitting process goes on.

The contribution is threefold: (i) a new reverse engineering framework to allow for the reconstruction of editable CAD models of both single parts and assemblies through a multi-step approach; (ii) a simulated annealing-based fitting technique to control the deviations between the geometries (e.g. 2D sketches, 3D features/parts and assemblies) and the point cloud, and to allow the simultaneous satisfaction of multi-dimensional constraints (section 4.3.1); (iii) a two-level filtering technique to capture and manage the boundaries of the geometries inside the overall point cloud and allow for local fitting and detection of the interfaces between the parts of the assemby. The advantages of the proposed approach are demonstrated on industrial examples, and a comparative study with commercial software is proposed.

The paper is organized as follows. Section 2 reviews the related works and positions the novel technique with respect to state-of-the-art approaches. The proposed framework is then introduced in section 3 and the details of the generic and multi-dimensional fitting technique are given in section 4. The approach is then tested and validated on several test cases (section 5). Section 6 ends the paper with conclusions and perspectives.

2 Prior works

Reverse engineering is playing a very important role in today's product development processes and in advanced industrial applications, where backward engineering is used to build a CAD model that is geometrically identical to the existing physical product or system [7, 8]. New low-cost data acquisition devices are emerging to digitize 3D objects with enhanced quality and reduced acquisition times. The 3D surface reconstruction of point cloud data has been a critical problem in the field of computer graphics and vision [2], and many scholars have done extensive research on it [10]. However, in most applications, surface reconstruction techniques end up generating dead or frozen models without any information of the constituting features and which cannot be used for product development processes especially when it comes to late design changes. Such reverse engineered models are difficult to modify because of the lack of control parameters. Recently, some attempts to move from traditional reverse engineering processes to knowledge-based techniques have also been suggested, in order to extract explicit and sharable knowledge embedded in the digital flow [11]. According to the type of prior knowledge exploited for the RE object reconstruction, the works in literature can be grouped into three main groups: using geometric primitive prior, global regularity prior, and the data driven prior.

The *geometric primitive* prior concerns the works related to the fitting of basic geometries in the given point cloud. Classical geometric approaches (point cloud segmentation and feature data extraction) can be considered under this prior, where surface primitives are reconstructed from the set of points obtained by an acquisition device. Many commercial software such as CATIA, RapidForm, Geomagic-Studio have adopted this approach. Several applications make use of surface recovery and have been the subject of several research works [12, 13, 14, 15, 16]. Traditional reverse engineering consists of the following steps: data acquisition, pre-processing (noise filtering and merging), triangulation, segmentation, and surface fitting to obtain CAD models [17]. Thus, sophisticated surface fitting algorithms are required to generate surfaces that accurately represent the 3D information described within the point cloud [8]. Among them, Sørensen et al. presents a method

Copyright © by ASME

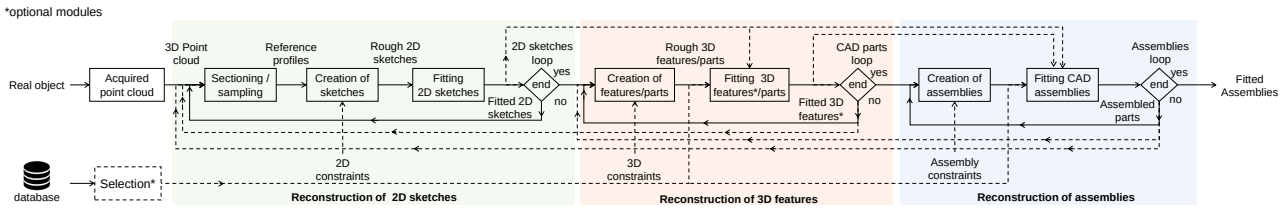

Fig. 1. Overall framework for the new reverse engineering technique exploiting a generic multi-dimensional fitting module.

based on a recursive RANSAC point removal approach, where an input point cloud is deconstructed into a set of convex point clusters, each corresponding to a mesh face of the final output model. The combined use of cluster analysis and RANSAC model extraction allows the method to operate on a relatively small number of possible vertex candidates for the final mesh, leading to an output mesh with relatively few faces compared to other approaches [18]. Azernikov et al. [19] have presented a fast approach for surface reconstruction from unorganized points. Their method is based on the extraction of a connectivity graph from the Hierarchical Space Decomposition Model (HSDM) and facet reconstruction. Using octree, reconstruction is performed in real-time. Such representation is suitable to process large-scale 3D data as the resulting mesh does not contain fine information on the constituting features (e.g fillets, chamfers). However, the resulting models are frozen meshes with no parametric features and assembly information, which make them hardly usable in the later stages of the PDP.

The *global regularity* prior deals with high-level properties such as symmetries, structural repetitions, and canonical relationships [20]. This is particularly interesting when considering the fitting of CAD models for which basic primitives can follow some specific rules. Among the existing techniques, Li et al. [21] have developed the so-called GlobFit method that simultaneously recovers a set of locally fitted primitives along with their global mutual relations. Starting with a set of initial RANSAC based locally fitted primitives [22], relations across the primitives such as orientation, placement and equality are progressively learned and conformed to. This algorithm operates under the assumption that the data corresponds to a man-made engineering object consisting of basic primitives, possibly repeated and globally aligned under common relations. Similarly, another approach by Lui et al. [23] extracts primitive shapes from point clouds using the RANSAC algorithm. Later, deviations of points from the fitted primitive shapes are analyzed by histograms. For point cloud patches segmented unreasonable, their approach updates parameters of segmentation according to the Gaussian noise and repeats the primitive shape detection process. After certain rounds of iteration, the approach can detect reasonable primitive shapes from point clouds. Their method is limited to very basic shapes. Monszpart et al. have proposed the so-called RAPter algorithm to abstract raw scans by regular arrangements of primitive planes by simultaneously extracting a set of primitives along with their inter-primitive relations [24]. However, those methods can hardly deal with CAD models made of more complex features, e.g. blends, draft features. Furthermore, they act at the level of the parts and not at the level of an assembly of parts. As they were initially designed to reconstruct basic primitives, this category of methods does not allow for a proper update of neither existing CAD models nor Digital Mock-Up (DMU). Such an ability would however be of great interest to update digital twins, and thus to answer Industry 4.0 needs.

The *data driven* prior uses existing objects to be fitted to point clouds in a rigid and non-rigid manner. Bey et al. have proposed a method to reconstruct 3D CAD models from point cloud data acquired in industrial environments, using a pre-existing 3D model as an initial estimate of the scene to be processed [25]. Most of the work is related to the reconstruction of cylindrical shapes using a greedy minimization method based on a stochastic exploration of the solution space. Buonamici et al. [26] have introduced a template-based technique for the reverse engineering of mechanical parts. Unfortunately, the method focuses on global fitting of parts, and neither the reconstruction nor update of CAD assembly models is considered. Erdos et al. [27] also worked on something similar for the reconstruction of industrial objects in a factory by adapting the shape of the CAD model to the given point cloud. Their work is also limited to simple shapes like cylinders. A similar work by Ishimtsev et al. [29] has introduced a method CAD-to-scan that fits by non-rigidly deforming retrieved CAD models. A non-rigid deformation model incorporating smooth transformations and preservation of sharp features, that simultaneously achieves very tight fits from CAD models to the 3D scan and maintains the clean, high-quality surface properties of hand-modeled CAD objects. Another approach is adopted by Durupt et al. to obtain a real CAD model with a tree structure of features called functional and structural skeleton [30]. Their semi-automated approach integrates classical geometric approach (point cloud segmentation and features data extraction) and a knowledge-based approach (functional and structural skeleton). However, their proposed work requires the interaction of multiple expertise to identify and classify the driving parameters. Thompson et al. [31] worked on a prototype of a reverse engineering system that uses manufacturing features as geometric primitives. The resulting models can be directly imported into feature-based CAD systems without loss of the semantics and topological information inherent in feature-based representations. The user specifies the types of manufacturing features present and the approximate location of each feature in the object. Their system does not yet deal with secondary feature properties such as chamfers, fillets,

Copyright © by ASME

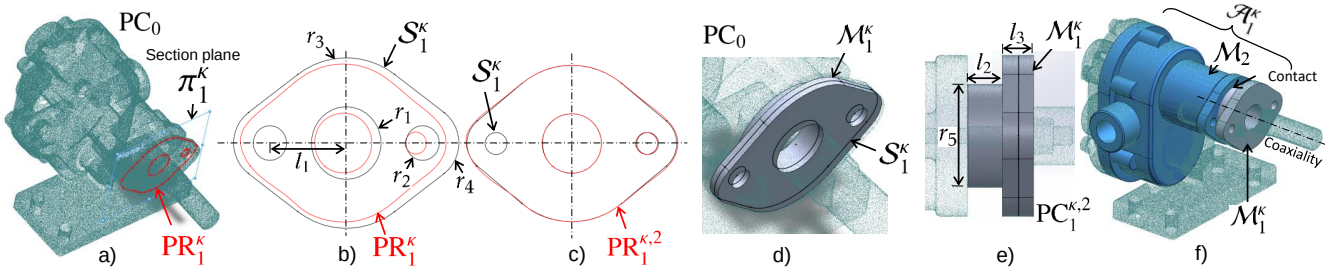

Fig. 2. Fitting a parameterized 2D sketch and 3D features in the point cloud of a digitized pump assembly: (a) sectioning with plane π_1^κ to extract the reference profile PR_1^κ in the initial point cloud PC_0; (b) drafting a rough parametric sketch $\mathcal{S}_1^\kappa$ in the vicinity of the reference profile PR_1^κ; (c) final 2D parametric sketch $\mathcal{S}_1^\kappa$ fitted in the filtered profile $PR_1^{\kappa,2}$; (d) draft CAD model $\mathcal{M}_1^\kappa$ extruded from the fitted 2D parametric sketch $\mathcal{S}_1^\kappa$ with an estimated extrusion length; (e) 3D CAD model $\mathcal{M}_1^\kappa$ fitted in the filtered point cloud $PC_1^{\kappa,2}$; (f) fitted assembly $\mathcal{A}_1^\kappa$ containing two models $\mathcal{M}_1^\kappa$ and $\mathcal{M}_2^\kappa$ assembled together with coaxiality and contact constraints.

and rounds and mostly single parts are considered.

As a conclusion, most of the existing methods are not yet able to efficiently reverse engineer a full DMU, i.e. an assembly of several parts constrained all together. Indeed, existing methods mostly work at the level of the parts and require the assembly to be disassembled prior to its digitalization. Moreover, the reconstructed CAD models are often considered as dead models that cannot be further edited in the later stages of the PDP, and this is a very limiting feature of the existing approaches.

3 Reverse Engineering framework

In our approach, instead of using a traditional patch-by-patch reconstruction process, the model reconstruction is obtained by adapting parameterized geometries to the corresponding point cloud, either locally or globally. Such geometries can be 2D sketches, 3D parts or assemblies' CAD models described in terms of their significant features, parameters, and constraints. Starting from an acquired point cloud PC_0, CAD models are reconstructed following the workflow depicted in Fig. 1. Depending on the available data, it admits different starting points. Nevertheless, it is up to the user to decide what are the most suitable steps to operate, and what are the parameters to be optimized. If a parameterized CAD template is not available, the user can start from scratch to obtain new 2D parametric sketches, well-matched with the given point cloud, usable to create the 3D part models. The user can come back to previous steps and the parameters are handled at the level of the sketches, parts, and assembly, all together or separately.

The framework supporting the new reverse engineering approach is composed of three modules, namely: (i) reconstruction of 2D parametric sketches; (ii) reconstruction of 3D features and parts; (iii) reconstruction of assemblies. All the modules use the same generic and multi-dimensional fitting algorithm which permits to adapt any parametric geometry to a given reference point cloud. The fitting process exploits a CAD kernel and a simulated annealing-based algorithm to solve the underlying numerical optimization problem. Depending on the reconstruction scenarios, the modules can be used independently or in combination with the others.

Given an acquired point cloud PC_0, the reverse engineering process follows several successive steps. At each step κ, with $\kappa \in [1..N_{step}]$ and N_{step} the number of steps required, a set $\mathbb{G}^\kappa$ of parametric geometries is created and used as input of the fitting algorithm. Each set $\mathbb{G}^\kappa = \{\mathcal{G}_i^\kappa,\ i \in [1..N_g^\kappa]\}$ contains N_g^κ geometries to be fitted at the same time, and each geometry $\mathcal{G}_i^\kappa$ can be a CAD model of a sketch, of a part or of an assembly. In the following sections, the three modules are first introduced separately, then section 3.4 discusses configurations where $\mathbb{G}^\kappa$ contains heterogeneous geometries fitted at the same time using the generic multi-dimensional fitting module presented in section 4.

3.1 Reconstruction of 2D parametric sketches

Linear and rotational sweep of 2D sections are classic operations when defining 3D features and parts. This module supports the reconstruction of 2D sketches fitting locally the acquired point cloud PC_0 at step κ of the reverse engineering process. Here, the set $\mathbb{G}^\kappa = \{\mathcal{S}_i^\kappa,\ i \in [1..N_s^\kappa]\}$ is composed of N_s^κ sketches to be fitted at the same time. Figures 2.a to 2.c illustrate the reconstruction of a single 2D sketch $\mathcal{S}_1^\kappa$. Generally speaking, each sketch $\mathcal{S}_i^\kappa$ lies in a user-specified section plane π_i^κ and is initially roughly defined and constrained. The numerical parameters of the sketch are variables for the fitting process that aims at automatically tuning their values to fit the related profile PR_i^κ (Fig. 2.b). Each profile is a polyline at the intersection between PC_0 and π_i^κ, and it can be considered as a 2D point cloud PC_i^κ (Fig. 2.a). Throughout the fitting process, profiles are cleaned up and further segmented by applying a two-step filtering strategy based on distance and normal computation. Therefore, the filtered profiles $PR_i^{\kappa,1}$ and $PR_i^{\kappa,2}$ are successively generated, and they can be considered as filtered 2D point clouds $PC_i^{\kappa,1}$ and $PC_i^{\kappa,2}$ issued from PC_0 (section 4.3.3). At each time, all the constraints are updated and managed by the CAD modeler, and the step κ ends up with a set of fitted sketches (Fig. 2.c).

Copyright © by ASME

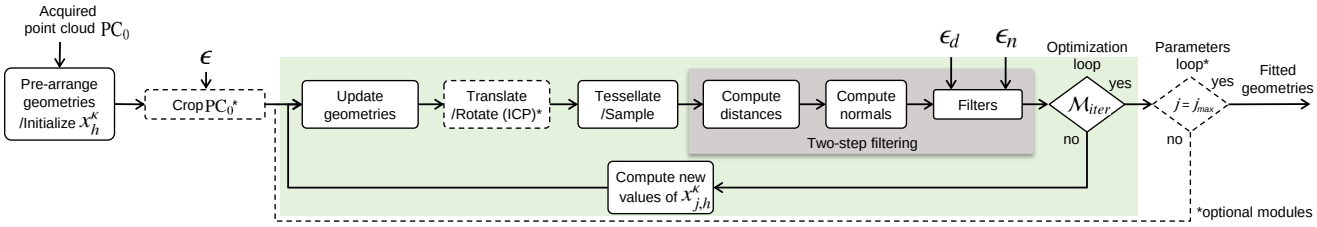

Fig. 3. Generic multi-dimentional fitting process handling parameters and constraints at the level of sketches, parts and assemblies.

3.2 Reconstruction of 3D features / CAD parts

This module aims at reconstructing CAD models using templates whose parameters have to be tuned to fit the point cloud, either locally or globally. Here, the set $\mathbb{G}^\kappa = \{\mathcal{M}_i^\kappa,\ i \in [1..N_m^\kappa]\}$ is composed of N_m^κ CAD models to be reversed at the same time. The templates can be specified following different scenarios (highlighted in orange in Fig. 1), either in sequence with the module described above or as a standalone process. The first scenario considers the use of a 2D sketch that has been previously fitted (Fig. 2.c). By extruding, revolving or using any other CAD operation on this 2D sketch, a draft CAD model $\mathcal{M}_i^\kappa$ is then created. At this stage, the parameters of the CAD operation can only be estimated (Fig. 2.d). This model can be further refined by adding additional features possibly obtained through the extrusion or revolution of other fitted sections. In another scenario, the template CAD model can directly come from a database or it can be roughly specified by the user. In this case, the user is required to pre-arrange the CAD model in PC_0. The fitting module then optimizes the shape of all the CAD models of $\mathbb{G}^\kappa$ to fit the point cloud (Fig. 2.e). At each optimization step, the point cloud is segmented using the two-level filtering strategy successively generating $PC_i^{\kappa,1}$ and $PC_i^{\kappa,2}$ for each model $\mathcal{M}_i^\kappa$ (section 4.3.3). This process is repeated for all the parts to be recovered (Fig. 2.f), either sequentially or simultaneously all together at the same time.

3.3 Reconstruction of assemblies

Assemblies can also be reconstructed and fitted by tuning their numerical parameters, e.g. the distance between two parts or the angle amongst faces of two adjacent parts. Here, the set $\mathbb{G}^\kappa = \{\mathcal{A}_i^\kappa,\ i \in [1..N_a^\kappa]\}$ is composed of N_a^κ assemblies or sub-assemblies. All the possible constraints the CAD modeler can handle (coincidence, perpendicularity, concentricity, parallelism, etc.) are managed, but only the numerical constraints e.g. angles and distances between faces, are used to adapt the CAD model to the point cloud during the optimization and the fitting process. Similarly, discrete constraints or boolean constraints, such as activation of user defined equations, are not managed within the optimization loop. The CAD modeler takes care of the additional assembly constraints (e.g. contact, perpendicularity) during the optimization process. Figure 2.f shows the reconstruction of an assembly of two models constrained by assembly constraints.

3.4 Reconstruction of heterogeneous geometries

One of the strengths of the proposed approach lies in the possibility to simultaneously reconstruct and fit heterogeneous geometries (i.e. 2D sketches, 3D parts and assemblies) in a single step κ. Indeed, heterogeneous geometries can be mixed up in a given set $\mathbb{G}^\kappa = \{\mathcal{G}_i^\kappa,\ i \in [1..N_g^\kappa]\}$ composed of N_g^κ geometries. For instance, fitting of an assembly may incorporate parameters of 2D sketches, 3D parts and sub-assemblies that are simultaneously updated for the reconstruction of an assembly model. Then, the fitting algorithm acts on the parameters (e.g. length, angle, radius, distance) while satisfying simultaneously all the constraints (e.g. parallelism, coaxiality) at different levels (2D sketch, part, assembly). Thus, even if the models are initially reversed sequentially, they can ultimately be further adjusted to perfectly fit the point cloud.

4 Generic multi-dimensional fitting module

This section details the module used to fit a set $\mathbb{G}^\kappa$ of N_g^κ geometries $\mathcal{G}_i^\kappa$ in a reference point cloud PC_0 at step κ of the reverse engineering process. This module is generic and works on geometries of any dimension, i.e. 2D sketches, single parts and assemblies. Figure 3 shows the different steps of the fitting module illustrated on the fitting of a gland in the point cloud of a digitized pump assembly (Fig. 2). Algorithm 1 provides the pseudo-code for the better understating of the developed method.

4.1 Pre-arrangement and cropping

Before fitting the N_g^κ geometries $\mathcal{G}_i^\kappa \in \mathbb{G}^\kappa$ in the point cloud, a pre-arrangement is required to bring the geometries near to the related point cloud portion and to initialize the values of its control parameters. This step results in the definition of a set $\mathbb{G}^\kappa(0)$ of pre-arranged geometries $\mathcal{G}_i^\kappa(0)$ at iteration step 0 of the optimization loop (Fig. 4.a). For sketches, the pre-arrangement is directly performed during their definition, being drafted over reference profiles obtained from the corresponding section planes. The same happens when the geometry is a CAD model obtained from an already fitted sketch (Fig. 2.c). As the fitted sketch is already aligned, the pre-arrangement consists in adjusting the parameters values of the adopted operator, for instance the length of the extrusion on the example of Fig. 2.d. The pre-arrangement is always necessary when the reference CAD model is taken from an existing database.

Copyright © by ASME

Algorithm 1

Input
Point cloud (PC_0)
Parametric CAD geometry ($\mathcal{G}_j^{\kappa}$)
Output
Fitted CAD geomerey ($\mathcal{G}_j^{\kappa}(i)$) in PC_0
Procedure
Pre-arrangement of CAD geometry with PC_0
Selection of parameters ($x_j^k(0)$) \\ initialization of parameter)
\\Cropping PC_0 (local fitting)
Parameters loop
  For $j = 1 : j_{max}$
    Optimization loop
    Repeat $i = 0 : M_{iter}$
      $\mathcal{G}_j^{\kappa}(i+1)$ = Update geometries ($x_j^k(0)$) \\ using CAD kernel
      Registration ($\mathcal{G}_j^{\kappa}(i)$, $PC_0(i)$)
      $\mathcal{G}_i^{\kappa\bullet}(i)$= Tessellation/sampling of geometry $\mathcal{G}_j^{\kappa}(i)$
      Two-step filtering
      Compute distances ($PC_0(i)$).$\mathcal{G}_j^{\kappa}(i)$)
      Compute normals & dot product ($PC_0(i)$).$\mathcal{G}_j^{\kappa}(i)$)
      $PC_0(i+1)$ = cropping $PC_0(i)$ \\using thresholds ε_d and ε_n
      $x_j^k(i+1)$ = Modifying geometries' parameters
    End
  End

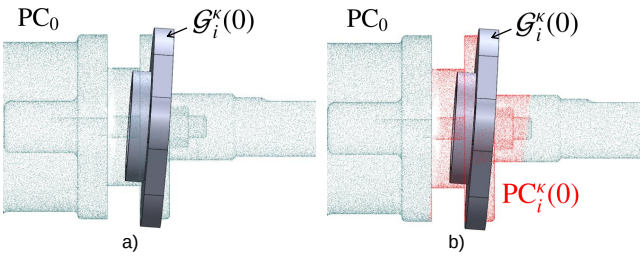

Fig. 4. Pre-arrangement of a single CAD model $\mathcal{G}_i^{\kappa}(0)$ in a reference point cloud PC_0 (a), and initial coarse cropping of PC_0 using a distance threshold to get the cropped point cloud $PC_i^{\kappa}(0)$ (b).

This initialization step also acts on PC_0 reducing the number of points to be considered when applying the two-level filtering strategy. The cropping operation is driven by an initial threshold ε that works on the distance between the points of PC_0 and the set $\mathbb{G}^{\kappa}(0)$ of pre-arranged geometries. Thus, for each geometry, all the points of PC_0 that have a distance to $\mathcal{G}_i^{\kappa}(0)$ greater than ε are cropped and the remaining points from $PC_i^{\kappa}(0)$ are used as input of the optimization process (Fig. 4.b). By default, ε is set to 10% of the diagonal of the considered pre-arranged geometrys bounding box. This value corresponds somehow to the level of confidence in the pre-arrangement.

4.2 Parameters loop

At each step κ of the reconstruction, and all along the fitting process, the algorithm creates instances of the geometries with different values of the chosen parameters. The control parameters of all the geometries $\mathcal{G}_i^{\kappa}$ can optionally be clustered in several groups G_j^{κ}, with $j \in [1..j_{max}]$, according to the level of details to which they correspond. Actually, grouping the parameters results in better and more stable fitting results and it also reduces the chances of getting the optimization process stuck in a local minimum. Considering all the parameters in a single group may result in under and

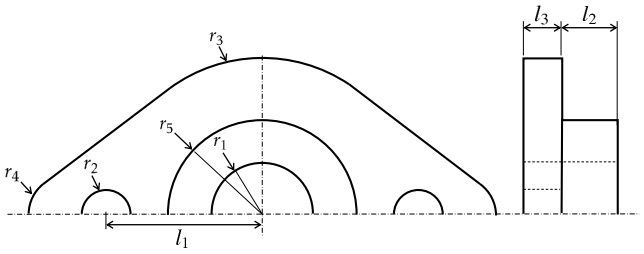

Fig. 5. Definition of the parameters controlling the gland of a pump: 5 parameters for the 2D sketch and 3 parameters for the 3D features.

overfitting of small features (e.g. fillets and chamfers). Parameters of group G_j^{κ} are denoted $p_{j,h}^{\kappa}$, $h \in \{1,...,N_{pj}^{\kappa}\}$ and are assigned to the numerical variables for the optimization loop, i.e. $x_{j,h}^{\kappa} = p_{j,h}^{\kappa}$. Using a three level decomposition turns out to be a good trade-off, considering the commonly adopted CAD modeling strategies (j_{max} = 3): G_1^{κ} groups the parameters of the assembly structure as well as the ones driving the structural features (e.g. pockets, revolutions), G_2^{κ} gathers the parameters of the detail features (e.g. holes, ribs) and G_3^{κ} includes the parameters used to finalize the shapes (e.g. fillets, chamfers). When considering the simultaneous fitting of several geometries, all the parameters are mixed up within those three groups. Thus, in the rest of the paper when referring to those groups, the lower index i of each geometry $\mathcal{G}_i^{\kappa}$ is not used.

It is important to understand that it is up to the user to decide which template and parameters are to be considered at step κ of the reverse engineering process. Figure 5 shows the parameterization of the gland whose fitting has already been introduced in Fig. 2. Overall, it is controlled by 8 control parameters ($N_p^{\kappa} = 8$): the radii r_i with $i \in [1..4]$, the length l_1, and three additional parameters r_5, l_2 and l_3 to control the volumetric (3D features) shape of the gland. Applying this grouping strategy, these parameters ($p_{j,h}^{\kappa}$) are divided into two groups as follows (upper index κ removed for sake of clarity):

$$\begin{aligned} G_1 = \{r_1, r_2, r_3, r_4, l_1\} :\ & p_{1,1} = r_1,\ p_{1,2} = r_2, \\ & p_{1,3} = r_3, p_{1,4} = r_4 \text{ and } p_{1,5} = l_1, \\ G_2 = \{r_5, l_2, l_3\} :\ & p_{2,1} = r_5\ ,\ p_{2,2} = l_2 \text{ and } p_{2,3} = l_3 \end{aligned} \tag{1}$$

The parameter loop is only activated if $j_{max} > 1$, otherwise the parameters are treated all together in a single group.

4.3 Optimization loop

The optimization loop is the core of the proposed approach. It is used at each step κ of the reverse engineering process, and for each group within the parameters loop. Starting from a set $\mathbb{G}^{\kappa}(0)$ of pre-arranged geometries $\mathcal{G}_i^{\kappa}(0)$, the algorithm iterates on the values $x_{j,h}^{\kappa}(t)$ belonging to the j-th group G_j^{κ}, thus creating an evolution of the N_g^{κ} geometries $\mathcal{G}_i^{\kappa}(t)$ until reaching a best fit of all the geometries in the filtered point clouds $PC_i^{\kappa,2}(t)$. Actually, this can

Copyright © by ASME

be formulated as a minimization problem:

$$\min_{\substack{x^{\kappa}_{j,h} \in \mathcal{D}^{\kappa}_{j,h} \\ h \in [1..N^{\kappa}_{pj}]}} E^{\kappa}_{j}(x^{\kappa}_{j,1}, ..., x^{\kappa}_{j,N^{\kappa}_{pj}}) = \sum_{i=1}^{N^{\kappa}_{g}} \mathrm{d}(\mathrm{PC}^{\kappa,2}_{i}, \mathcal{G}^{\kappa}_{i}) \qquad (2)$$

where $x^{\kappa}_{j,h}$ are the variables, $\mathcal{D}^{\kappa}_{j,h}$ their definition domains, and E^{κ}_{j} the energy function characterizing the overall deviation for the complete set $\mathbb{G}^{\kappa}$ at the j-th step of the parameters loop. The filtering technique is explained in section 4.3.3, and the way distances between the filtered $\mathrm{PC}^{\kappa,2}_{i}$ and the geometry $\mathcal{G}^{\kappa}_{i}$ are computed is explained in section 4.3.4. Throughout the optimisation process, the built-in constraints (e.g. symmetry, parallelism) amongst the geometries are managed by a CAD modeler also in charge of keeping their consistency.

4.3.1 Simulated Annealing algorithm

To optimize the parameters of the geometries $\mathcal{G}^{\kappa}_{i} \in \mathbb{G}^{\kappa}$ in order to fit the filtered point clouds $\mathrm{PC}^{\kappa,2}_{i}$, the reconstruction process requires an algorithm that can solve the optimization problem in a large solution space. Research and experimentation carried out brought to the choice of the simulated annealing (SA) algorithm. Other metaheuristics, such as Particle Swarm Optimization (PSO), have been tested but have demonstrated a lower efficiency than SA. This is because PSO has two main search capabilities, the exploration and exploitation, whereas SA only performs exploitation. Thus, in the context of this work, PSO does require more updates of the CAD modeler than SA. Indeed, in the carried out experiments on the fitting of 2D sketches, the results show that the PSO takes about 10 times more iterations than SA, thus an average of 10 times more updates of the geometric models causing a drastic decrease in the overall fitting process. Moreover, both optimization strategies may result in configurations that are hard for the CAD modeler to update, and may even cause the software to crash. But, SA performs better and with this approach very few issues are encountered when updating the geometries during the fitting process. SA was designed for stochastic search problem that successfully avoids local minimum during the search process. It uses a probabilistic approach to move from one point to another in search of global optima. This transition process of finding the optimal solution in a large search space depends on the temperature and the change in the objective function. SA optimization algorithm mimics the physical process of heating a material and then slowly reducing its temperature that decreases the overall energy of the system (annealing) to remove the defects in the material. SA-based fitting of CAD models also works on the same principle where the overall energy (sum of the distances between the geometries $\mathcal{G}^{\kappa}_{i}$ and the filtered point cloud $\mathrm{PC}^{\kappa,2}_{i}$) is minimized as shown in Eqn. (2). The algorithm handles a limited set of constraints, mainly the lower and upper bounds of the values, and both the internal and assembly constraints (e.g. coincident, parallelism) are directly handled by the CAD modeler in charge of the updates. This is particularly interesting as the geometric constraints are satisfied using black boxes combining calls to procedures of the CAD modeler [32,33]. For optimal results, the SA algorithm initial temperature T_0 needs to be tuned. From the experiences, the initial temperature is set to 10 in most of the reconstruction cases. In some cases, if the fitting results are not satisfactory, the user can re-run the optimization process with a different initial temperature value.

4.3.2 Update, translate/rotate, tessellate and sample

Before running the two-level filtering technique required for local fitting, the geometries taking part to the current optimization loop need to be prepared. Thus, the N^{κ}_{g} geometries involved at the κ-th step of the reverse engineering process are successively updated, translated and rotated, tessellated and sampled as the fitting process goes on, i.e. at each iteration t of the optimization loop. Each geometry $\mathcal{G}^{\kappa}_{i}(t)$ is thus updated by the CAD modeler according to the evolution of the parameters values $x^{\kappa}_{j,h}(t)$. Here, updates can appear at the level of the sketches, parts and assemblies. Then, each geometry is translated and rotated using an ICP algorithm [34] that finds the best fit rigid body transformation between the geometry and its associated $\mathrm{PC}^{\kappa}_{i}(t)$. Indeed, adding six additional parameters to control the position and orientation of each geometry taking part to the optimization process would clearly reduce the performances of the SA algorithm, it is therefore more efficient to manage these issues in a standalone ICP step. However, ICP is optional and should not be run when an assembly has been updated and the assembly constraints are already satisfied. Finally, two additional geometric representations are generated, successively $\mathcal{G}^{\kappa\triangle}_{i}(t)$ after the tessellation and $\mathcal{G}^{\kappa\bullet}_{i}(t)$ after the sampling. The sampling is performed so as to keep same the density of both the sampled geometry $\mathcal{G}^{\kappa\bullet}_{i}(t)$ and its corresponding cropped point cloud $\mathrm{PC}^{\kappa}_{i}(t)$. In case of a 2D sketch, the cropped point cloud is a reference profile $\mathrm{PR}^{\kappa}_{i}(t)$ that is also sampled.

4.3.3 Two-level point cloud filtering technique

When considering local fitting, the reference point cloud needs to be processed and segmented at each iteration t of the optimization loop so as to remove points that are not meaningful for the reverse engineering of geometry $\mathcal{G}^{\kappa}_{i}(t)$. This is performed using a two-level filtering technique, which analyses the deviations in terms of distance and normal evolution between the point cloud and the sampled geometry to be fitted. This technique also provides a segmentation of the point cloud to allow local fitting. Figure 6 illustrates this filtering process on the example of a square-shaped 2D sketch $\mathcal{S}^{\kappa}_{i}(t)$ to be fitted to a reference profile $\mathrm{PR}^{\kappa}_{i}(t)$ presenting a protrusion not to be considered during the local fitting. At an iteration step t of the optimization loop, the two-level filtering strategy works as follows:

1. Find the matching between points of the sampled geometry $\mathcal{G}^{\kappa\bullet}_{i}(t)$ and the ones of the reference point cloud $\mathrm{PC}^{\kappa}_{i}(t)$, respectively $\mathcal{S}^{\kappa\bullet}_{i}(t)$ and $\mathrm{PR}^{\kappa}_{i}(t)$ on the 2D ex-

Copyright © by ASME

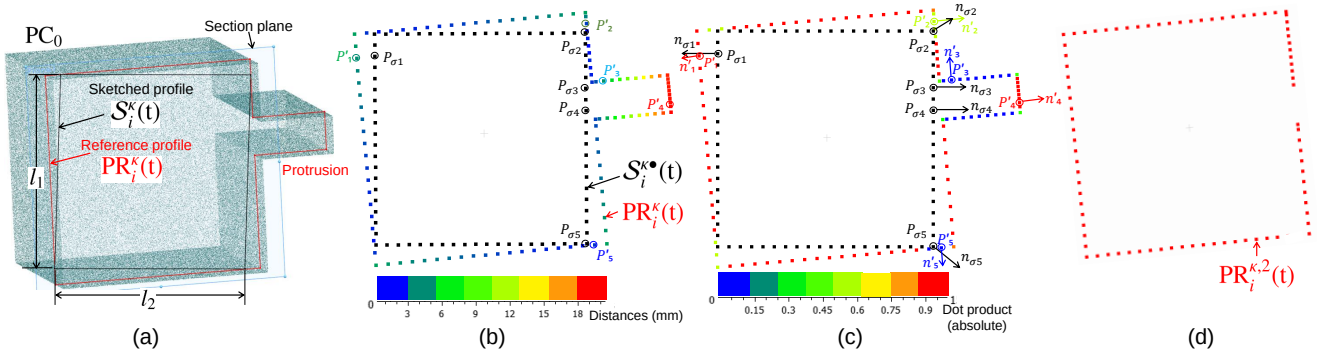

Fig. 6. The two-level filtering technique applied on a square-shaped 2D sketch: (a) parameterized sketch $\mathcal{S}_i^{\kappa}(t)$ close to the reference profile $\mathrm{PR}_i^{\kappa}(t)$ with a protrusion not to be considered during the fitting; (b) points of $\mathcal{S}_i^{\kappa}(t)$ and $\mathrm{PR}_i^{\kappa}(t)$ after sampling, with colors showing the distances between the sampled sketch and profile; (c) same points as in (b) but with colors showing the deviations in the orientation of the normals between the sampled sketch and profile; (d) two-level filtered profile $\mathrm{PR}_i^{\kappa,2}(t)$ defining the point cloud used for the computation of the energy.

ample of Fig. 6.b. This is performed while computing the distance field between the points of the reference point cloud and the ones of the geometry. The output of this step is a list of matched pairs of points $\mathcal{L}_d = \{(P'_k; P_{\sigma(k)}), k \in [1..N'_p]\}$ with N'_p the number of points of the reference point cloud to be filtered. Here, $\sigma(k)$ returns the index of the closest point of P'_k in the sampled geometry, with $\sigma(k) \in [1..N_p]$ and N_p the number of points in the sampled geometry. Depending on the geometric configuration, $\sigma(k)$ can return the same index for different values of k.

2. Filter the point cloud while removing points P'_k (with $k \in [1..N'_p]$) that do not satisfy to the two conditions:

$$\begin{aligned} \|P'_k P_{\sigma(k)}\| &< \varepsilon_d \\ |n'_k \cdot n_{\sigma(k)}| &> \varepsilon_n \end{aligned} \tag{3}$$

where n'_k and $n_{\sigma(k)}$ stand for the unit normals at points P'_k and $P_{\sigma(k)}$ respectively, and $\cdot$ returns the dot product of the two vectors. Using the threshold ε_d, a filtered point cloud $\mathrm{PC}_i^{\kappa,1}(t)$ is first generated while removing points whose distance to the geometry is too large. Then, the result is further processed using the threshold ε_n to get $\mathrm{PC}_i^{\kappa,2}(t)$ while removing points whose normal deviates too much with respect to the normal at the matching point.

The way the points of the reference point cloud behave with respect to those two conditions is illustrated on Figures 6.b and 6.c, whereas Fig. 6.d shows the result of the two-level filtering only keeping points required for the energy computation.

4.3.4 End of the optimization loop

Once the geometries sampled and filtered, at each iteration t of the optimization loop, the energy function involved in Eqn. (2) is computed using the following function:

$$\mathrm{d}(\mathrm{PC}_i^{\kappa,2}(t), \mathcal{G}_i^{\kappa}(t)) = \sum_{\tau=1}^{n_i^{\bullet}(t)} \mathrm{d}^2(\mathcal{G}_i^{\kappa\bullet}(t)[\tau], \mathrm{PC}_i^{\kappa,2}(t)) \tag{4}$$

where $n_i^{\bullet}(t)$ is the number of points in $\mathcal{G}_i^{\kappa\bullet}(t)$. The function $\mathrm{d}(point, cloud)$ returns the nearest neighbor distance for each point in the $cloud$. The optimization loop stops when a max number of iteration M_{iter} without change of E_j^{κ} (up to an accuracy ε_{iter}) is reached, otherwise the SA algorithm goes on and computates new values $x_{j,h}^{\kappa}(t+1)$ defining a new set $\mathbb{G}^{\kappa}(t+1)$ of geometries $\mathcal{G}_i^{\kappa}(t+1)$ used as new inputs of the optimization loop.

5 Results and discussion

This section illustrates the novel reverse engineering technique on two industrial assemblies. The core of the approach has been implemented in MATLAB® that handles the SA algorithm, computes the distances and normals, and also filters the point cloud. It calls the built-in functions of SolidWorks® to perform the successive updates of the geometries, to tesselate the models and to ensure the consistency of the resulting 2D sketches, B-Rep models and assemblies during the optimization loops. The sampling and the ICP algorithm are run in CloudCompare also called in batch mode.

5.1 Reconstruction of the Wankel engine assembly using as-scanned point cloud

The first example aims at reverse engineering multiple parts of a Wankel engine (Fig. 7.a). The reference point cloud PC_0 is composed of 667k points obtained using an as-scanned point cloud generation strategy [35], which uses the Hidden Point Removal (HPR) algorithm from 12 viewpoints [36]. Before generating the point cloud, the CAD assembly was simplified while removing and remounting some

Copyright © by ASME

components so as to capture its internal details (Fig. 7.b). Using an as-scanned point cloud, the real values of the parameters are known from the original CAD models, and the accuracy of the proposed method can be easily evaluated. Being p_h^F the final parameter values of the fitted CAD models, and p_h^D the original ones as they appear in the DMU, the relative deviations δp_h and the absolute deviations Δp_h can be computed as follows:

$$\Delta p_h = \frac{|\delta p_h|}{|p_h^D|} = \frac{|p_h^D - p_h^F|}{|p_h^D|} \; \forall h \in [1..N_p] \tag{5}$$

where N_p stands for the overall number of parameters possibly distributed in the parameter groups. The deviations of the parameters are listed in Tab. 1.

Table 1. Results for the partial reverse engineering of the Wankel engine (steps $\kappa = 1$ to 9).

κ	Groups	p_h	p_h^0 (mm)	p_h^D (mm)	p_h^F (mm)	δp_h (mm)	Δp_h
Piston							
1	G_1	r_1	190	195	198.78	-3.7800	0.0194
		l_1	200	180.15	180.67	-0.5200	0.0029
	G_2	r_2	63	55	54.9375	0.0625	0.0011
	G_3	r_3	3	5	4.8841	0.1159	0.0232
		l_2	95	95	94.7978	0.2022	0.0021
2	G_1 (3D)	el_1	75	80	78.6038	1.3962	0.0175
Casing							
3	G_1	r_4	120	129.69	128.38	1.3100	0.0101
		r_5	80	92	90.84	1.1600	0.0126
		r_6	20	92	22.294	69.7060	0.7577
		l_3	60	55.75	48.214	7.5360	0.1352
	G_2	α_6	130	129.5	129.39	0.1100	0.0008
		α_7	166	166	166.19	-0.1900	0.0011
		of_1	7	9.6	9.5088	0.0912	0.0095
		r_7	4	6.5	6.4252	0.0748	0.0115
		r_8	110	111.6	111.15	0.4500	0.0040
		r_9	13	11	11.983	-0.9830	0.0894
	G_3	α_1	29.5	31	30.837	0.1630	0.0053
		α_2	49	48.24	48.223	0.0170	0.0004
		α_3	64	64.77	64.746	0.0240	0.0004
		α_4	80	80.92	80.89	0.0300	0.0004
		α_5	106	104.96	104.99	-0.0300	0.0003
		r_{10}	2	2.5	1.764	0.7360	0.2944
4	$G_1(3D)$	el_2	70	80	79.2869	0.7131	0.0089
End cover							
5	G_1	r_8	111	111.6	111.44	0.1600	0.0014
		r_{12}	12	9	10.652	-1.6520	0.1836
		α_8	100	99	100.2	-1.2000	0.0121
		r_{13}	61	62	60.981	1.0190	0.0164
		r_{14}	2	3.2	2.7616	0.4384	0.1370
		α_9	60	60	58.321	1.6790	0.0280
		r_{15}	68	72.5	72.224	0.2760	0.0038
	G_2	c_3	20	21.63	21.4764	0.1531	0.0071
6	G_1 (3D)	el_3	26	35	35.0725	-0.0725	0.0021
7	G_1 (3D)	el_4	5	2.5	0.9126	1.5874	0.6350
Cover plate							
8	G_1 (3D)	el_5	10	7	7.0764	-0.0764	0.0191
Fillet							
9	G_1 (3D)	$fillet$	2	5	6.7520	-1.750	0.3500

The reverse engineering process follows a part-by-part reconstruction strategy where parts are constrained all together with assembly constraints (e.g. coincidence of axes, contact between faces). Here, due to space limitation, only the reconstruction of the first 5 parts is shown for a total of 9 reverse engineering steps: piston ($\kappa \in \{1,2\}$), casing ($\kappa \in \{3,4\}$), end cover ($\kappa \in \{5,6,7\}$), cover plate ($\kappa = 8$) and filleting ($\kappa = 9$). Since the engine has symmetries, additional CAD operators are also used.

The piston is reconstructed at first following a two-step procedure (Figures 7.e$_1$ to 7.e$_4$): fitting of the sketch profile ($\kappa = 1$) and fitting of the extrusion length ($\kappa = 2$). The reference profile is obtained while sectioning the initial point cloud PC_0 with the section plane π^1 (Fig. 7.c). The resulting reference profile PR^1 is composed of sub-profiles that are managed by the two-level filtering technique. To reconstruct the piston, a rough parametric sketch $\mathcal{S}^1$ is drafted around the reference profile related to the piston only (Fig. 7.e$_1$). ICP is also used to handle the alignment and orientation of the updated geometry throughout the optimization process. The five parameters controlling the shape of the piston are divided into three groups. Parameters r_1 and l_1 are associated to G_1 and thus fitted first, followed by r_2 for the central circle. Three small holes at the tips of the piston are controlled by r_3 and l_2 associated to G_3 and thus fitted last to obtain the fully fitted sketch (Fig. 7.e$_2$). The fitted profile is then extruded with a coarse value of the length el_1 to generate the volumetric shape of the piston (Fig. 7.e$_3$). To get the final value of the extruded length el_1, an initial cropping is automatically performed on PC_0 before starting the optimization. The resulting red points in Fig. 7.e$_3$ still contain some unwanted points not belonging to the shape of the piston. Thus, all along the optimization process, these points are filtered and removed using the two-level filters until obtaining the segmented blue points (Fig. 7.e$_4$). Table 1 shows very good fitting results for the two corresponding steps ($\kappa \in \{1,2\}$). Finally, as there are two pistons in the engine, the one fitted first is then duplicated (Fig. 7.e$_4$).

The next component to be reversed is the casing following a two-step procedure (Figures 7.f$_1$ to 7.f$_4$): fitting of the sketch profile ($\kappa = 3$) and fitting of the extrusion length ($\kappa = 4$). This part is a bit complicated because it has many parameters to be optimized. The reference profile PR^1 is the same as the one obtained at iteration $\kappa = 1$. Here again, a draft sketch $\mathcal{S}^3$ is first created with 16 parameters to be optimized (Figures 7.f$_1$ and 7.f$_2$) and clustered in 3 groups according to the level of detail they refer to (Tab. 1). Figure 8 shows the evolutions of the parameters of each group for the reconstruction of the casing section. The group G_1 is optimized first, then G_2, and G_3 at last. It can be seen that parameters are evolving a lot in the beginning as the required solutions are away from the desired one. Later, the curves get smoother as the values of deviation of parameters decreases. Actually, the largest deviations in the G_1 is for the parameter r_6. This is due to the fact that the radius is rather small and its initial value is far away from the desired one and very few points can contribute to the fitting. The stochastic nature of the SA algorithm is clearly visible. Such an algo-

Copyright © by ASME

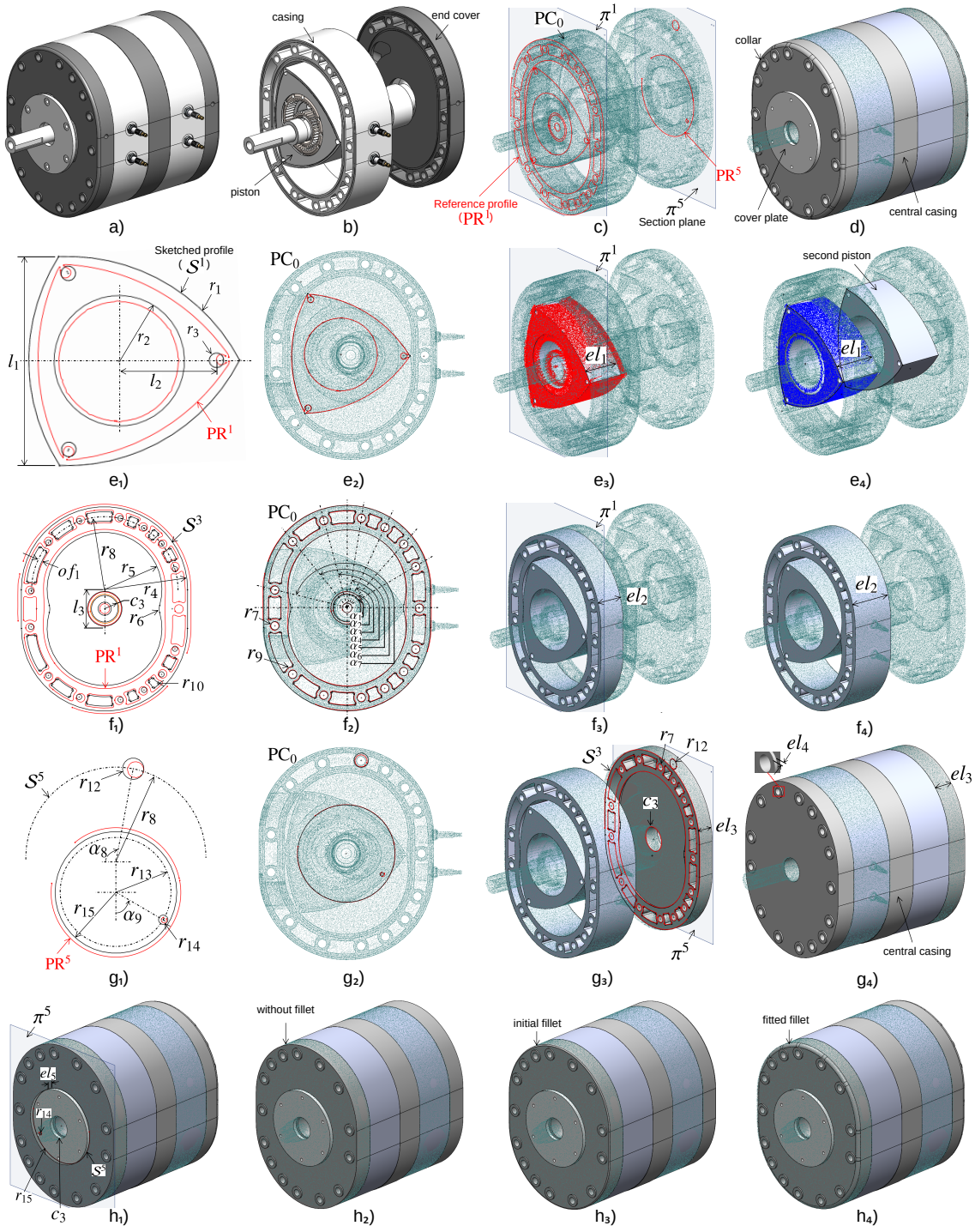

Fig. 7. Successive steps of the reverse engineering of a Wankel engine with a point cloud generated from an as-scanned virtual scanning.

rithm is interesting to solve this global optimization problem in a large search space. Thus, splitting the parameters into several groups can improve the efficiency of the overall fitting framework. After recovering the section, the extrusion length el_2 can be further optimized ($\kappa = 4$) using the fitting procedure from a coarse extrusion (Figures 7.f$_3$ and 7.f$_4$).

Then, the end cover is reverse engineered in three steps (Figures 7.g$_1$ to 7.g$_4$): fitting of the sketch profile ($\kappa = 5$), fitting of the main extrusion ($\kappa = 6$) and fitting of the collar ($\kappa = 7$). First, the reference profile PR^5 is obtained while sectioning PC_0 with the plane π^5 (Fig. 7.c), and then used to fit a draft sketch $\mathcal{S}^5$ (Figures 7.g$_1$ and 7.g$_2$). Here, only two groups have been used. Once fitted, some elements of sketch $\mathcal{S}^5$ are combined with elements of sketch $\mathcal{S}^3$ and they are used to create the draft extrusion of length el_3. This length is then optimized in step $\kappa = 6$ (Fig. 7.g$_3$). The collar is then recovered while optimizing its extrusion length el_4 in step $\kappa = 7$ and the resulting feature is repeated and aligned

Copyright © by ASME

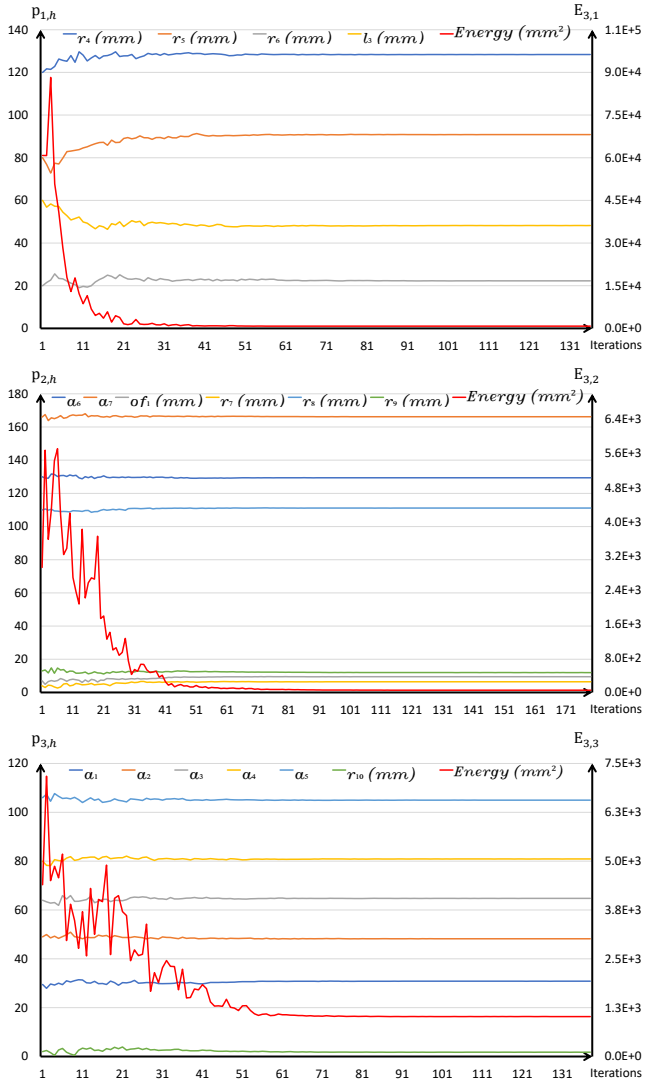

Fig. 8. Simulated annealing evolution curves for the three parameters groups to reconstruct the parametric sketch of the casing part.

with the surrounding holes of sketch $\mathcal{S}^3$. Finally, the casing and the end cover are duplicated and the central casing created in between using geometric entities extracted from the previously fitted sketches (Fig. 7.g$_4$).

The penultimate step consists in reconstructing the cover plate using elements of the previously fitted sketches $\mathcal{S}^3$ and $\mathcal{S}^5$. Here, only its extrusion length el_5 is to be fitted at step $\kappa = 8$ (Figures 7.h$_1$ and 7.h$_2$).

Finally, a fillet is added to the end cover, and its radius is optimized at step $\kappa = 9$ (Figures 7.h$_3$ and 7.h$_4$). All the optimized values are reported in Tab. 1. Of course, for certain reverse engineering scenarios, the values p_h^F obtained at the end of the reverse engineering process can be rounded.

5.2 Reconstruction of a hydraulic rotary actuator using real scanned point cloud

The second example validates the proposed reverse engineering approach on the real scanned point cloud of a hydraulic rotary actuator (Fig. 9.a). The point cloud has been acquired using a ROMER Absolute Arm 7520 SI and while scanning the assembly as a whole, i.e. without disassembling the whole assembly prior to its digitalization. To capture internal details of assembly some components are removed before scanning. For instance, out of four screws, two screws are removed from the both sides of assembly to capture the information of internal diameters r_2 and r_3. For the cylinder part, the external diameter r_1 is retrieved using our approach while internal diameter is assumed by pipe's standard thickness. The raw point cloud has been pre-processed before entering the reconstruction steps (i.e. noise and outliers removal, registration, filtering). By the end of this step, a clean point cloud PC_0 having about one million points is obtained (Fig. 9.b). Here, as the CAD models are not available to serve as a ground truth, the accuracy of the approach is evaluated while comparing the fitted values p_h^F to the ones p_h^D measured directly on the parts using a caliper (Tab. 2).

Table 2. Results for the partial reverse engineering of a hydraulic rotary actuator (steps $\kappa = 1$ to 9).

κ	Groups	p_h	p_h^0 (mm)	p_h^D (mm)	p_h^F (mm)	δp_h (mm)	Δp_h
Central casing							
1	G_1	l_1	98	93.42	93.4532	-0.0332	0.0004
		l_2	47	45.90	45.8171	0.0829	0.0018
		l_3	94	90.40	93.3780	-2.9780	0.0329
		l_4	32.5	32.00	32.7104	-0.7104	0.0222
		l_5	13	15.10	14.7835	0.3165	0.0210
		l_6	7.5	7.50	7.4345	0.0655	0.0087
		l_7	20	17.00	17.5090	-0.5090	0.0299
	G_2	l_8	25	24.35	23.9505	0.3995	0.0164
		l_9	17	17.25	17.4360	-0.1860	0.0108
		l_{10}	36	32.70	36.9264	-4.2264	0.1292
		l_{11}	26	25.70	25.5856	0.1144	0.0045
		r	3	8.50	4.3074	4.1926	0.4932
2	$G_1(3D)$	el_1	84	88.06	88.5479	-0.4879	0.0055
Cylinder and screws							
3	G_1	r_1	36	37.60	37.6544	-0.0544	0.0014
		r_2	42	42.20	42.3345	-0.1345	0.0032
		r_3	5	3.95	3.9943	-0.0443	0.0112
4	$G_1(3D)$	el_2	100	108	110.5972	-2.5972	0.0240
Right end part							
5	G_1	er_1	44	42.20	42.5169	-0.3169	0.0075
		er_2	38	39.50	39.6997	-0.1997	0.0051
		er_3	10	15.00	13.9397	1.0603	0.0707
		er_4	9	7.80	7.8498	-0.0498	0.0064
6	$G_1(3D)$	el_3	10	15.85	15	0.8500	0.0536
Screw							
7	G_1	br_1	7.5	6.40	6.4042	-0.0042	0.0007
8	$G_1(3D)$	el_4	2	5.50	5.4772	0.0228	0.0041
Sub-assembly							
9	G_1	er_2	39.6997	39.50	38.7445	0.7555	0.0191
		r_3	3.9943	3.95	3.832	0.1180	0.0299
		el_3	15	15.85	15.7958	0.0542	0.0034
		br_1	6.4042	6.40	6.4257	-0.0257	0.0040
		el_4	5.4772	5.50	6.1002	-0.6002	0.1091
		r_1	37.6544	37.60	37.6267	-0.0267	0.0007
		r_2	42.5169	42.20	42.5599	-0.3599	0.0085
		el_2	107	108.00	108.2916	-0.2916	0.0027

Due to space limitation, only the reconstruction of the

Copyright © by ASME

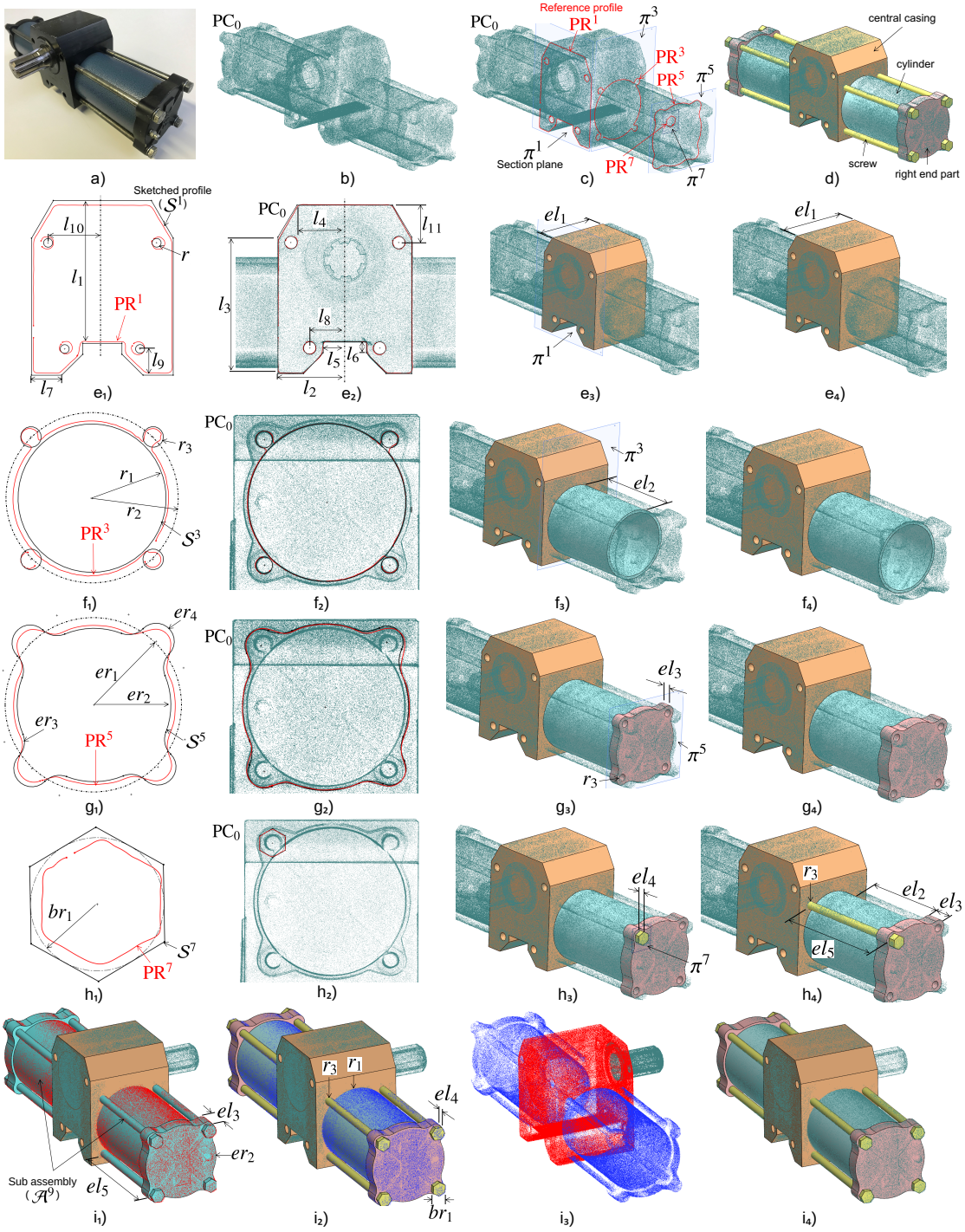

Fig. 9. Successive steps of the reconstruction of a hydraulic rotary actuator based on a point cloud obtained using a laser scanner without disassembling the parts.

first 4 parts are presented for a total of 9 reverse engineering steps (Fig. 9.d): central casing ($\kappa \in \{1,2\}$), cylinder ($\kappa \in \{3,4\}$), right end part ($\kappa \in \{5,6\}$), screw ($\kappa \in \{7,8\}$) as well as a final overall fitting of the assembly ($\kappa = 9$). At each reconstruction step, the fitting is performed using the two-level filtering technique, in order to clearly segment the point clouds and identify parts and interfaces. Figure 9.c shows the reference profiles of the four parts extracted using section planes. Section plane π^1 is defined first using a fitting on user-specified points located on the front face of the central casing, and it serves as a reference to locate the three other planes.

The central casing is reconstructed first following a two-step procedure (Figures 9.e$_1$ to 9.e$_4$): fitting of the sketch profile ($\kappa = 1$) and fitting of the extrusion length ($\kappa = 2$). Thus, a rough parametric sketch $\mathcal{S}^1$ is drafted using 12 control parameters gathered together in two groups G_1 and G_2 (Figures 9.e$_1$ and 9.e$_2$). The resulting values p_h^F after the opti-

Copyright © by ASME

mization are listed in Tab. 2. Then, at step $\kappa = 2$, the extrusion length el_1 is optimized using SA algorithm from a user-specified coarse value (Figures 9.e$_3$ and 9.e$_4$). The fitted central casing now serves as a reference to reconstruct the three other parts, using assembly constraints handled by the CAD modeler.

The next component to be reverse engineered is the cylinder following a two-step procedure: (Figures 9.f$_1$ to 9.f$_4$): fitting of the sketch profile ($\kappa = 3$) and fitting of the extrusion length ($\kappa = 4$). Thus, at step $\kappa = 3$, a rough sketch $\mathcal{S}^3$ is drafted around the reference profile PR^3 using 3 control parameters gathered together in a single group (Figures 9.f$_1$ and 9.f$_2$). Here, the two parameters r_2 and r_3 appear at this step even if they define respectively the positioning and diameter of the four fixation screws to be reconstructed later in the reverse engineering process. Thus, at step $\kappa = 4$, only r_1 is used for the extrusion whose length el_2 is optimized using SA algorithm from a user-specified coarse value (Figures 9.f$_3$ and 9.f$_4$). As the point cloud has been obtained without disassembling the parts, the thickness of the cylinder cannot be captured directly and has to be specified by the user.

Then, at step $\kappa = 5$, the right end part is reconstructed from reference profile PR^5 created with section plane π^5 that is constrained to be parallel to the lateral face of the casing. A draft sketch $\mathcal{S}^5$ defined by 4 control parameters taking part to a single group is fitted to the reference profile (Figures 9.g$_1$ and 9.g$_2$). This sketch is then enriched with the geometric entities of $\mathcal{S}^3$ parameterized by r_2 and r_3. Doing this way, the complete sketch can be extruded of a coarse length el_3 that is finally tuned using the SA algorithm (Figures 9.g$_3$ and 9.g$_4$). As for all the other steps, the two-level filtering technique allowed to distinguish points belonging to the right end part, from points belonging to the cylinder and screws.

Finally, at step $\kappa = 7$, screws are reconstructed using the section plane π^7 to extract the reference profile PR^7 of the screw head. The drafted parametric sketch $\mathcal{S}^7$ is controlled by parameter br_1 whose best fitting value is obtained using the SA algorithm (Figures 9.h$_1$ and 9.h$_2$). The screw head is generated while optimizing its extrusion length el_4 (Fig. 9.h$_3$), and the bottom part of the screw is directly reconstructed using an extrusion of length $el_5 = el_2 + el_3$ (Fig. 9.h$_4$). During the fitting, the screw is constrained to stay on the circle of radius r_2. Moreover, screws being standard parts, the obtained values for screws can be normalized according to those available from catalogs. As for the thickness of the cylinder, the length of the screw inside the central casing cannot be retrieved thus it has to be adjusted afterwards.

To complete the full reconstruction of the CAD assembly, the screw is duplicated using circular pattern around the radius r_2. Then, except the central casing, all the reconstructed parts are duplicated and mirrored so as to get the symmetric side of the actuator (Fig. 9.i$_1$). All those parts are gathered together in a sub-assembly $\mathcal{A}^9$ whose parameters are further optimized in a last reconstruction step ($\kappa = 9$). Actually, the idea is to exploit the rest of the point cloud, i.e. the one part that has not yet been fitted to any template, to adjust the parameters values of the geometries already fitted on the right end side. Here, all the duplicated geometries take part to the optimization process and the only parameters that are further tuned are listed in Tab. 2. The average absolute deviation between the point cloud and the obtained CAD sub-assembly is about 0.0377 mm (min = -2.6173 mm, max = 6.0638 mm, std = 0.5474 mm) which is also quite low. The assembly also has internal constraints (e.g. contact between faces, coaxiality, relationships between parameters notably between the left and right parts) directly handled by the CAD modeler during the successive updates. An additional constraint that was not yet taken into account has been added and consists in imposing the equality $r_2 = er_1$. Figure 9.i$_4$ shows the result of the final fitted assembly, and Fig. 9.i$_3$ the segmented point cloud distinguishing points belonging to the fitted sub-assembly (blue) from the ones of the central casing (red).

5.3 Interface in SolidWorks

The prototype software has been integrated as a plugin in SolidWorks®2017 Education Edition. The developed user interface allows to select the optimization technique and set the required parameters, as shown in Fig. 10. For the selection of the optimization strategy, the user has to specify the type of geometry to fit: a *2D curve* or *3D part(s)/assembly*. Depending on the choice, the optimization pipeline will choose its path either for the 2D profile fitting or 3D part fitting. Once the optimization strategy has been selected, the input values of the parameters of the optimization algorithm (e.g. function tolerance, maximum iterations, stall iteration limit, initial temperature for SA) have to be specified. Default values are presented that can be set and modified by the user. Then, the free parameters of the CAD geometry have to be specified, possibly together with their lower and upper bounds.

Finally, in the result section the user can select different options related to the parameters computed during the optimization for the evaluation of the process. Thus, for instance, the user can get the initial value of the energy function at starting point after pre-arranging the CAD part or 2D sketch to the reference point cloud. Additional information on the evolution of energy of parameters and the total time can also be saved. Once every section has been set, the user can execute the fitting by pressing the Run Optimization button.

5.4 Comparison with commercial software

To show the strengths and weaknesses of the proposed approach, it has been compared with two commercial software: CATIA V5 and DesignX. Table 3 highlights some features of our approach in comparison with the two mentioned software.

Traditional software for RE mostly address the reconstruction of patches that are then trimmed and stitched together to obtain a closed boundary surface geometry. This geometry is then transformed into a volumetric shape to obtain the corresponding CAD model. Differently, the method proposed in this paper does not require the creation and merging of the bounding patches, even if it supports such ca-

Copyright © by ASME

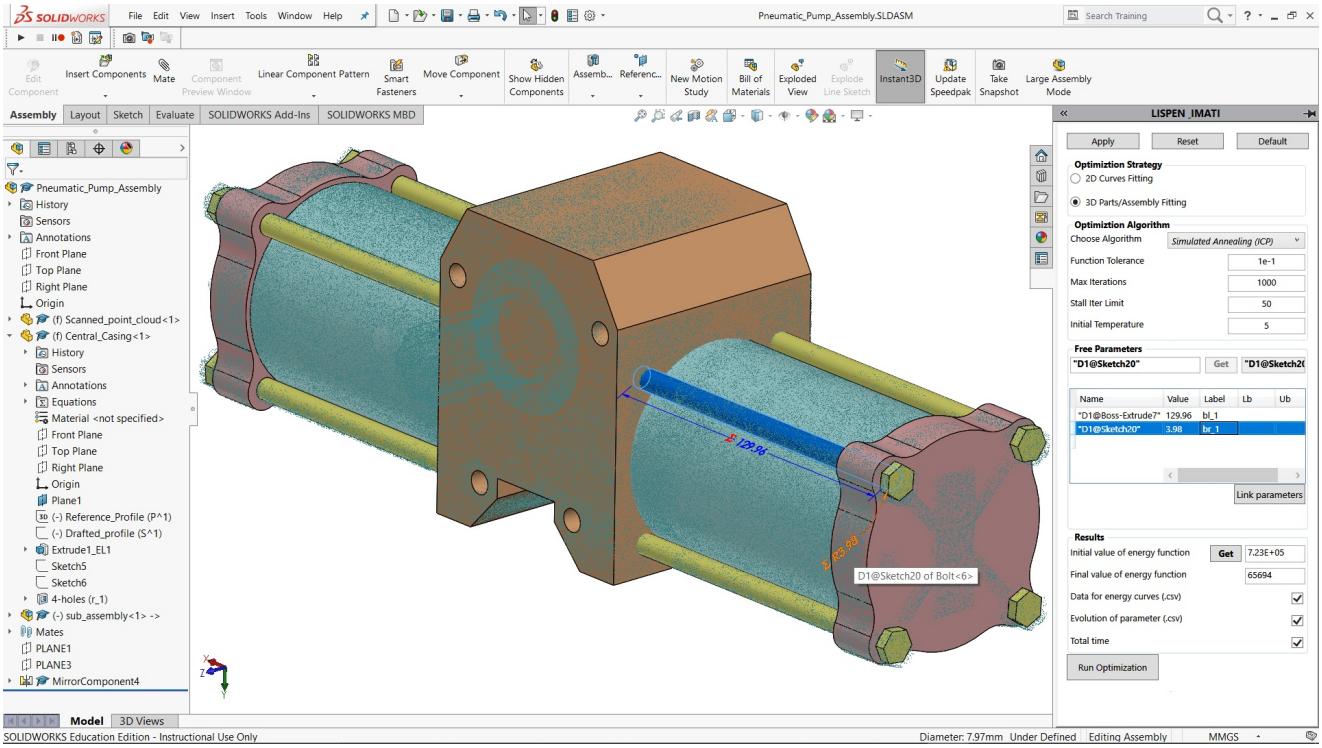

Fig. 10. Optimizer interface in SolidWorks for reconstructing draft CAD models to be used all along the PDP.

Table 3. Comparison of RE with CATIA, DesignX and with our method.

Criterion	New RE	CATIA	DesignX
Patch-by-patch reconstruction	✓	✓	✓
Feature-by-feature reconstruction	✓	-	Partial
Part-by-part reconstruction	✓	-	-
Assembly fitting	✓	-	-
Robust to noise and outliers	✓	-	-
Constraints satisfaction	✓	Difficult	Difficult
Segmentation	Auto	Manual	Auto
Editable part reconstruction	✓	-	Not always
Robust with partial point clouds	✓	Difficult	Difficult
Registration	✓	✓	✓
Late design changes	✓	-	-

pability. Patches can be produced from the template CAD geometries in the form of surfaces that are offset from the faces and then optimized for fitting using distance constraints. Features are not recognized in CATIA, while DesignX allows the reconstruction of a limited number of features like revolves, extrusion, etc. In the proposed method, the resulting CAD model directly includes the composing features in the case they are present in the adopted template. Small features, like fillets and chamfers, can be reconstructed even at end of the optimization process in the case they are not present in the template, therefore not considered in the early stage of the optimization. In addition, the method allows a feature-by-feature model reconstruction, allowing local fitting of sections and surfaces, the user can iteratively recreate the CAD model proving the composing feature templates.

Amongst the main strengths of the proposed method is the capability of dealing with the reconstruction of complete assemblies, using either a part-by-part or directly a whole assembly CAD template. This capability is missed in the existing traditional RE software like CATIA and DesignX. This is also due to the provided segmentation capabilities that are crucial to identify the boundaries of parts within an assembly point cloud. This process is manual in CATIA, where users are supported with tools like the brush to segment set of points. DesignX allows to automatically segment and identify the different sets of points to generate surfaces. In the proposed method, segmentation is automatically performed within the optimization loop using the two-level filtering.

Constraint satisfaction also plays an important role during the reconstruction process. In traditional RE, if constraints, like concentricity, parallelism and perpendicularity, are not maintained properly during the reconstruction process, they can be hardly addressed at the end to correct the obtained model. Following our RE technique, the CAD modeler guarantees their maintenance and satisfaction throughout the optimization process.

In CATIA at the end of the RE process, frozen B-Rep CAD models are obtained, which are not editable CAD and therefore can hardly support further subsequent design changes of their key features. On the other hand, DesignX can deliver CAD models in which some selected features are editable, like revolution, extrusion, holes, etc; however, this possibility is limited when the part is complex with many features. Moreover, while to fully obtain an editable CAD

Copyright © by ASME

model with DesignX requires a certain skill, the here presented method directly adapts the features during the optimization process. Another strong point of this method is that the user can come back anytime to any step during the reconstruction process. Thus, if not fully satisfied of the obtained results, any reconstructed part, used as a reference for other parts, as well as multiple reconstructed parts can be further fitted at the end of the optimization process. For instance, the user can roughly fit the components of an assembly first locally, and then he/she can perform a global fitting with higher accuracy. At this level, the global fitting can make use of selected parameters of sketches, 3D parts, and assembly features simultaneously. This is not possible in CATIA and DesignX as the chosen references at the beginning of the reconstruction process will go all along the way in the traditional RE and do not allow any modification at the end of the process.

6 Conclusion and future works

This paper has introduced a novel framework for the reconstruction of parametric CAD models sketches, parts and assemblies from point clouds. The proposed approach bypasses the traditional reverse engineering technique where surfaces are reconstructed patch-by-patch to create dead CAD models that cannot be exploited later in the PDP. Here, both local and global fitting can be performed, considering geometries either one-by-one, or all together to further optimize the fitting afterwards. This is particularly interesting to avoid disassembling parts to be scanned individually. The resulting models can be edited as their geometry is driven by multiple control parameters. The strengths of the approach lie in the definition and use of a multi-dimensional fitting module able to optimize the shape of various geometries (i.e. 2D sketches, 3D parts and assemblies) so as to fit to an input point cloud, while also capturing and managing their boundaries and the interfaces inside the overall point cloud. This has been made possible thanks to a new two-level filtering technique able to segment the point cloud and remove points that should not be taken into account within the optimization loops. During the fitting process, the geometries are updated by a CAD modeler, which also takes care of the internal constraints used to maintain the consistency of the models, and also allows higher-level specifications (e.g. coxiality between axes, contact between faces, relationships between parameters). Moving from a patch-by-patch to a part-by-part reconstruction strategy, users do not have to solve time-consuming trimming or stitching issues. The approach has proved to be accurate and it has been tested on virtually generated scans as well as on point clouds obtained from a real acquisition device.

However, it is still up to the users to make certain decisions like what are the reconstruction steps, or what are the variables to be used and how to group them within the optimization loops. This will be further investigated in the future. To support this decision process, the use of machine learning techniques will be studied. Indeed, using such techniques on numerous results obtained following varied reconstruction paths, it will be possible to correlate the followed strategies to different quality metrics, e.g. errors on the parameters values, overall deviation in terms of distance between the points and models, or number of steps required to accomplish the reconstruction [37].

References

[1] Lu, Y., 2017. "Industry 4.0: a survey on technologies, applications and open research issues". *Jou. of Industrial Information Integration,* **6**, pp. 1–10.

[2] Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., and Silva, C. T., 2017. "A survey of surface reconstruction from point clouds". In Computer Graphics Forum, Vol. 36, Wiley Online Library, pp. 301–329.

[3] Falcidieno, B., Giannini, F., Léon, J.-C., and Pernot, J.-P., 2014. "Processing free form objects within a product development process framework". *Advances in Computers and Information in Engineering Research*, pp. 317–344.

[4] Buonamici, F., Carfagni, M., Furferi, R., Governi, L., Lapini, A., and Volpe, Y., 2018. "Reverse engineering modeling methods and tools: a survey". *Computer-Aided Design and Applications,* **15**(3), pp. 443–464.

[5] Sommer, M., Stjepandić, J., Stobrawa, S., and von Soden, M., 2019. "Automatic generation of digital twin based on scanning and object recognition". In *Transdisciplinary Engineering for Complex Socio-technical Systems*. IOS Press, pp. 645–654.

[6] Shabayek, A. E. R., Aouada, D., Cherenkova, K., and Gusev, G., 2020. "Towards automatic CAD modeling from 3D scan sketch based representation". In Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020), GRAPP, pp. 392–398.

[7] Dúbravčík, M., and Kender, Š., 2012. "Application of reverse engineering techniques in mechanics system services". *Procedia Engineering,* **48**, pp. 96–104.

[8] Shabani, B., and Pandilov, Z., 2017. "Analyzing and application of reverse engineering for design and development of mechanical parts". *Mechanical Engineering - Scientific Journal,* **35**(2), pp. 89–96.

[9] De Toledo, R., Levy, B., and Paul, J.-C., 2008. "Reverse engineering for industrial-environment CAD models". In International Symposium on Tools and Methods of Competitive Engineering, TMCE 2008.

[10] Guo, B., Wang, J., Jiang, X., Li, C., Su, B., Cui, Z., Sun, Y., and Yang, C., 2020. "A 3D surface reconstruction method for large-scale point cloud data". *Mathematical Problems in Engineering,* **2020**.

[11] Anwer, N., and Mathieu, L., 2016. "From reverse engineering to shape engineering in mechanical design". *CIRP Annals,* **65**(1), pp. 165–168.

[12] Petitjean, S., 2002. "A survey of methods for recovering quadrics in triangle meshes". *ACM Computing Surveys (CSUR),* **34**(2), pp. 211–262.

Copyright © by ASME

[13] Lukács, G., Marshall, A., and Martin, R., 1997. "Geometric least-squares fitting of spheres, cylinders, cones and tori". *RECCAD, Deliverable Document 2 and 3, COPERNICUS project*(1068).

[14] Werghi, N., Fisher, R., Ashbrook, A., and Robertson, C., 1999. "Faithful recovering of quadric surfaces from 3d range data". In 2nd Int. Conf. on 3-D Digital Imaging and Modeling (Cat. No. PR00062), pp. 280–289.

[15] Satish, A., Rambabu, K., and Ramji, M., 2013. "Part modelling with reverse engineering". *Int. J. of Eng. Resch. & Tech,* **2**, pp. 486–94.

[16] Mohaghegh, K., Sadeghi, M., and Abdullah, A., 2007. "Reverse engineering of turbine blades based on design intent". *The International Journal of Advanced Manufacturing Technology,* **32**(9-10), pp. 1009–1020.

[17] Kumar, A., Jain, P., and Pathak, P., 2013. "Reverse engineering in product manufacturing: an overview". *DAAAM Int. Scientific Book*, pp. 665–678.

[18] Sørensen, T. A., Mark, N., and Møgelmose, A., 2021. "A ransac based CAD mesh reconstruction method using point clustering for mesh connectivity". In International Conference on Machine Vision and Applications (ICMVA).

[19] Azernikov, S., and Fischer, A., 2004. "Efficient surface reconstruction method for distributed CAD". *Computer-Aided Design,* **36**(9), pp. 799–808.

[20] Mitra, N., Wand, M., Zhang, H. R., Cohen-Or, D., Kim, V., and Huang, Q.-X., 2013. "Struct-aware shape process". *SIGGRAPH Asia 2013 Courses*, pp. 1–20.

[21] Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J., 2011. "Globfit: Consistently fitting primitives by discovering global relations". *ACM Transactions on Graphics,* **30**(4), pp. 1–12.

[22] Schnabel, R., Wahl, R., and Klein, R., 2007. "Efficient ransac for point-cloud shape detection". *Computer Graphics Forum,* **26**(2), pp. 214–226.

[23] Liu, J., 2020. "An adaptive process of reverse engineering from point clouds to CAD models". *International Journal of Computer Integrated Manufacturing,* **33**(9), pp. 840–858.

[24] Monszpart, A., Mellado, N., Brostow, G. J., and Mitra, N. J., 2015. "Rapter : Rebuilding man-made scenes with regular arrangements of planes". *ACM Trans. Graph.,* **34**(4), pp. 1–12.

[25] Bey, A., Chaine, R., Marc, R., Thibault, G., and Akkouche, S., 2011. "Reconstruction of consistent 3D CAD models from point cloud data using a priori CAD models". In ISPRS workshop on laser scanning, Vol. 1.

[26] Buonamici, F., Carfagni, M., Furferi, R., Governi, L., Lapini, A., and Volpe, Y., 2018. "Reverse engineering of mechanical parts: A template-based approach". *Journal of Computational Design and Engineering,* **5**(2), pp. 145–159.

[27] Erdős, G., Nakano, T., and Váncza, J., 2014. "Adapting cad models of complex engineering objects to measured point cloud data". *CIRP Annals,* **63**(1), pp. 157–160.

[28] Benko, P., Kós, G., Várady, T., Andor, L., and Martin, R., 2002. "Constrained fitting in reverse engineering". *Computer Aided Geometric Design,* **19**(3), pp. 173–205.

[29] Ishimtsev, V., Bokhovkin, A., Artemov, A., Ignatyev, S., Niessner, M., Zorin, D., and Burnaev, E., 2020. "Cad-deform: Deformable fitting of cad models to 3d scans". In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, Springer, pp. 599–628.

[30] Durupt, A., Remy, S., and Ducellier, G., 2011. "Reverse engineering of a piston using knowledge based reverse engineering approach". In *Global Product Development*. Springer, pp. 683–690.

[31] Thompson, W. B., Owen, J. C., Germain, H. d. S., Stark, S. R., and Henderson, T. C., 1999. "Feature-based reverse engineering of mechanical parts". *IEEE Transactions on robotics and automation,* **15**(1), pp. 57–66.

[32] Gouaty, G., Fang, L., Michelucci, D., Daniel, M., Pernot, J.-P., Raffin, R., Lanquetin, S., and Neveu, M., 2016. "Variational geometric modeling with black box constraints and dags". *Computer-Aided Design,* **75**, pp. 1–12.

[33] Pernot, J.-P., Michelucci, D., Daniel, M., and Foufou, S., 2019. "Towards a better integration of modelers and black box constraint solvers within the product design process". *Annals of Mathematics and Artificial Intelligence,* **85**(2), Apr, pp. 147–173.

[34] Besl, P. J., and McKay, N. D., 1992. "A method for registration of 3-D shapes". *IEEE Trans. Pattern Anal. Mach. Intell.,* **14**(2), Feb., pp. 239–256.

[35] Montlahuc, J., Shah, G. A., Polette, A., and Pernot, J.-P., 2018. "As-scanned point clouds generation for virtual reverse engineering of cad assembly models". *Computer-Aided Design and Applications*, pp. 1–11.

[36] Katz, S., Tal, A., and Basri, R., 2007. "Direct visibility of point sets". *ACM Transactions on Graphics,* **26**(3), pp. 1–11.

[37] Danglade, F., Pernot, J.-P., Véron, P., and Fine, L., 2017. "A priori evaluation of simulation models preparation processes using artificial intelligence techniques". *Computers in Industry,* **91**, pp. 45–61.

Copyright © by ASME