

ÉCOLE DOCTORALE SCIENCES ET MÉTIERS DE L'INGÉNIEUR
[LISPEN - Campus d'Aix-en-Provence]

THESIS

to obtain the degree of doctor issued by

École Nationale Supérieure d'Arts et Métiers

Speciality : Conception (AM)

and

Università degli Studi di Genova

Curriculum meccanica, misure e robotica (ciclo XXXIII)

presented by : Ghazanfar Ali SHAH

defended on : July 1st, 2021

**Template-based reverse engineering of parametric CAD
models from point clouds**

Directors of thesis:

[PERNOT Jean-Philippe, GIANNINI Franca]

Co-director of thesis:

[POLETTE Arnaud and MONTI Marina]

Research reference at DIME for the PhD programme:

[ZOPPI Matteo]

Jury

M. Nabil ANWER

Professor, LURPA, Paris-Sud University

Examiner

M. Giovanni BERSELLI

Professor, DIME, Università degli studi di Genova

Examiner

Mme. Caterina RIZZI

Professor, DIGIP, Università degli Studi di Bergamo

Reviewer

M. Jean-Luc MARI

Professor, LIS, Aix-Marseille University

Reviewer

M. Jean-Philippe PERNOT

Professor, LISPEN, Arts et Métiers

Examiner

M. Arnaud POLETTE

Ass.Professor, LISPEN, Arts et Métiers

Examiner

Mme. Franca GIANNINI

Research director, IMATI-CNR Genova

Examiner

Mme. Marina MONTI

Senior researcher, IMATI-CNR Genova

Examiner

Dedicated to my Mom and Dad,
who always supported and encouraged me to go on every adventure,
especially this one.

Acknowledgments

The present thesis was developed in the framework of a co-tutoring agreement between the Università degli Studi di Genova and the École Nationale Supérieure d'Arts et Métiers (campus of Aix-en-Provence). The financial support of the Higher Education Commission (HEC) of Pakistan is gratefully acknowledged and for its continued commitment to enhance the quality of higher education in Pakistan. The research has been carried out in Ingénierie des Systèmes Physiques et Numériques Laboratory (LISPEN Laboratory – EA 7515) and in the Istituto di Matematica Applicata e Tecnologie Informatiche (IMATI) Consiglio Nazionale delle Ricerche (CNR) in Genova.

First and foremost, I would like to express my sincere gratitude to all my thesis supervisors and co-supervisors for their invaluable guidance and encouragement throughout my Ph.D. The nervousness and hesitation for the time duration in Italy turned out to be a lifetime experience with excitement and motivation under the supervision of Mdm Franca GIANNINI and Marina MONTI. They not only guided me in my research activities but also helped me with my tiniest issues during my stay in Italy and made me feel like part of the family. I can not thank enough my two supervisors in France, Jean-Philippe PERNOT and Arnaud POLETTE who have been my supervisors since the beginning of my Master's internship and whose long and fruitful discussions have provided me many helpful suggestions, important pieces of advice, and constant encouragement throughout this research time. In particular, a special thanks is due to Prof. Jean-Philippe PERNOT, who would deserve another manuscript only for acknowledgments. At many stages in the course of this research project, I benefited from his advice, particularly so when exploring new ideas. His positive outlook and confidence in my research inspired me and gave me confidence. His careful follow-ups and meeting contributed enormously to the production of this thesis. I have been extremely lucky to have him as my supervisor who cared so much about my work and myself.

ACKNOWLEDGMENTS

I would also like to sincerely thank all the jury members, especially, Prof. Nabil ANWER and Prof. Giovanni BERSELLI for accepting of being the examiners of the PhD committee. I am extremely grateful to Prof. Caterina RIZZI and Prof. Jean-Luc MARI for accepting to be the reviewers and for their valuable time and effort in reviewing the thesis.

I would like to thank every member of the IMATI and LISPEN institutes over the years I worked with you, not only for sharing scientific ideas during my work but also for the kindness and friendly time we have spent together. Among them, I have found not only colleges but friends to share life with. I'm sure we will keep in touch for a long while.

Lastly, I would like to give the most heartfelt thanks to my family, my father, my brothers, my friends in Aix-En-Provence and my wife for their selfless support, deep love, and endless encouragement. Their meticulous care and excellent support are the key sources of my continuous endeavor, progress, and growth in both study and life.

Sincerely,

Ghazanfar Ali SHAH

Table of contents

Acknowledgments	v
List of Tables	x
List of Figures	xvi
Abbreviation	xvii
Résumé	xix
Abstract	xxi
Sommario	xxiii
Publications	xxv
Introduction	1
1 As-scanned point clouds generation for virtual Reverse Engineering of CAD assembly models	11
1.1 Introduction	12
1.2 Conclusion	27
2 Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies	29
2.1 Introduction	30

TABLE OF CONTENTS

2.2	Conclusion	52
3	Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications	53
3.1	Introduction	54
3.2	Conclusion	68
4	User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models	69
4.1	Introduction	70
4.2	Conclusion	85
	Conclusion & perspectives	87
	Bibliographie	93
A	Annex	95
A.1	Developed prototype	96
A.1.1	Optimization strategy	97
A.1.2	Optimization algorithm	97
A.1.3	Selection of free parameters	98
A.1.4	Results	99
A.2	Application of the prototype plugin	100
A.2.1	Selection and prearrangement of the CAD model	100
A.2.2	Setting optimizer (plugin values)	101
A.2.2.1	Optimization strategy	101
A.2.2.2	Free parameters	102
A.2.2.3	Results	102

List of Tables

1	Connections of the four chapters with the scientific contributions.	8
1	As-scanned point clouds generation for virtual Reverse Engineering of CAD assembly models	
1	Characteristics of the modules with respect to their ability to incorporate artifacts appearing when scanning real-life objects.	18
2	Parameters of four configurations characterizing different scanning sessions to be simulated using the proposed virtual Reverse Engineering framework.	21
3	Results of the virtual Reverse Engineering process applied on three first CAD models (sonotrode, compressor, engine) and following three different scanning configurations.	21
4	Results of the virtual Reverse Engineering process applied on CAD models according to the fourth scanning configuration of Table 2.	24
5	Default values of the framework's parameters so that the generated as-scanned point clouds fit the ones obtained by a ROMER Absolute Arm 7520 SI.	24
2	Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies	
1	Results for the global fitting of a L-like shape (without noise and with an initial temperature of the SA algorithm $T_0 = 10$).	43
2	Evolution of the relative deviation Δp_k with respect to the amplitude of the inserted noise ($0 \mu\text{m}$, $50 \mu\text{m}$, $100 \mu\text{m}$) for the global fitting of the L-like shape.	44
3	Results for the global fitting of a Sonotrode to a real point cloud obtained by a laser scanner.	45

LIST OF TABLES

4	Results obtained when fitting three successive moves of robot arms parameterized by three rotation angles.	46
5	Results for the local fitting of multiple parts in a valve assembly.	49
6	Absolute and relative errors measured between the point cloud and each locally fitted part of the valve assembly.	50
3	Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications	
1	Values of the setting parameters split in two categories and gathered together in a so-called configuration \mathcal{C}_κ used for fitting the rectangular flange.	58
2	Results for the global fitting of rectangular flange following the configuration \mathcal{C}_κ of Table 1.	59
3	List of setting parameters with corresponding three levels ($level_{1-3}$) used for setting up the DoEs.	61
4	Fitting of rectangular flange for 18 configurations \mathcal{C}_κ ($\kappa \in [1..18]$) of DoEs using L_{18} table of Taguchi orthogonal array.	62
5	Grouping of parameters p_k according to their sensitivity levels and to the types of feature they refer to.	64
6	Similarity check for the bend part with models in the database.	66
4	User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models	
1	Results for the partial reverse engineering of the Wankel engine (steps $\kappa = 1$ to 9).	79
2	Results for the partial reverse engineering of a hydraulic rotary actuator (steps $\kappa = 1$ to 9).	82
3	Comparison of RE with CATIA, DesignX and with our method.	91

List of Figures

A1	From data acquisition of an object to the CAD model.	2
A2	Workflow for the traditional reverse engineering process, and possible difficulties. . . .	4
A3	Overall framework for the RE to reconstruct mechanical parts/assembly models. . . .	6
1	As-scanned point clouds generation for virtual Reverse Engineering of CAD assembly models	
1	As-scanned point cloud generation supporting the fitting and new RE technique to reconstruct mechanical parts/assembly models.	12
1	Real sonotrode (a1) scanned to get point clouds that have been post-treated (a2). CAD model of a sonotrode (b1) virtually reverse engineered following three parameterizations of the proposed framework (b2-b4).	16
2	Different artifacts: (a) non-uniform sampling, (b) missing data, (c) misaligned scans, (d) noisy data and (e) outliers.	17
3	Virtual Reverse Engineering modular framework generating as-scanned point clouds from CAD assembly models.	17
4	Sampling module: (a) watertight triangle mesh, (b) $densitySamp = 2.5$, (c) $densitySamp = 5$	18
5	Hidden Point Removal module: (a) initial point cloud, (b) visible points after the HPR operator running from a given viewpoint shown with a cone, (c) from two viewpoints without merge ($Merging = 0$).	19
6	Misalignment module with three viewpoints, the barycenter as a rotation center and with the $maxAngle = 3.5$	20

LIST OF FIGURES

7	Noise module moving points along the normal to the surface (<i>dirNoise</i>): (a) zoom window, (b) <i>levelNoise</i> = 0, (c) <i>levelNoise</i> = 0.03, (d) <i>levelNoise</i> = 0.1	20
8	CAD model of a compressor reversed engineered with configurations 1 to 3.	22
9	CAD model of a RC engine reversed engineered with configurations 1 to 3.	23
10	Results of the virtual Reverse Engineering process applied on CAD models following configure 4 of Table 2, from left to right and top to bottom: clutch plate, connecting rod, clutch plate assembly, radial engine rod, hand wheel, wheel assembly, vice, machine element and stop valve assembly.	23
2	Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies	
1	Fitting technique supporting the new reverse engineering to reconstruct mechanical parts/assembly models.	30
1	Overall framework to fit parameterized CAD assembly models to point clouds of digitized mechanical part assemblies.	36
2	Fitting the parameterized CAD model of a flange in the point cloud of a digitized valve assembly ($N_m = 1$): a coarse pre-arrangement in the initial point cloud PC_0 ; b–d evolution of the cropped point clouds PC_i during the segmentation loops; e final fitted flange $\mathcal{M}_{i_{max}}$ and associated segmentation $PC_{i_{max}}$	38
3	Definition of the eight parameters controlling the shape of the half-flange that is fitted in the point cloud of a digitized assembly as depicted on Fig. 2.	39
4	Generation of as-scanned point clouds: a CAD assembly model to be virtually scanned, b point cloud without noise, and with noise, c using different view points.	41
5	Global fitting of a L-like shape following three optimization loops: a coarse pre-arrangement in the initial point cloud; b loop on \mathcal{G}_1 ; c loop on \mathcal{G}_2 ; d final fitted part after a loop on \mathcal{G}_3	42
6	Simulated annealing evolution curves for the three parameters loops ($j \in \{1, \dots, 3\}$) of the last segmentation step ($i = 3$) during the global fitting of a L-like shape: parameters values evolution (left) and energy evolution (right)	43

LIST OF FIGURES

7 Global fitting of a Sonotrode to a real scanned point cloud following 3 optimization loops: **a** scanned Sonotrode; **b** coarse pre-arrangement in the initial point cloud; **c** loop on \mathcal{G}_1 ; **d** loop on \mathcal{G}_2 ; **e** final fitted part after a loop on \mathcal{G}_3 44

8 Fitting of robot arms constrained with assembly constraints and parameterized by 3 rotation angles: **a** initial configuration; **b–d** final configurations fitting respectively the first, second and third robot moves. 45

9 Successive fittings of multiple CAD models in the point cloud of a digitized valve assembly: (a_1) coarse pre-arrangement of two flanges in the initial point cloud; (a_2) final fitted flanges; (b_1) pre-arrangement of two screws; (b_2) final fitted screws; (c_1) pre-arrangement of two nuts; (c_2) final fitted nuts; (d_1) pre-arrangement of the central part; (d_2) final fitted central part; (e_1) pre-arrangement of the top plate; (e_2) final fitted top plate; (f_1) pre-arrangement of the bottom plate; (f_2) final fitted bottom plate; (g_1 to g_4) successive segmentations of PC_0 48

3 Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications

1 Optimization of the Simulated Annealing-based fitting and the new reverse engineering technique for efficient reconstruction of the mechanical parts/assembly models. 54

1 Fitting of a CAD assembly on the point cloud of a digitized pneumatic pump: coarse result without tuning the parameters of the SA-based fitting algorithm (a); and good-quality result after tuning (b). 55

2 Global fitting of a rectangular flange defined by 8 control parameters: (a) initial configuration after the prearrangement step, (b) fitted part after 50 iterations of the SA algorithm, (c) final fitted part minimizing the deviation with the point cloud ($\epsilon_s=10^{-1}$ and $M_{iter}=1000$). 57

3 Overall framework used to populate a database of CAD models stored together with their best configurations \mathcal{C}_κ for the SA. 59

4 Changes of $\delta p_{k,j}$ imposed by the SA algorithm at each iteration j of the fitting process on the rectangular flange. 59

LIST OF FIGURES

5	Sensitivities $\mathcal{S}_{k,0}$ of the 8 parameters p_k of the rectangular flange at prearranged position ($j = 0$).	60
6	Grouping parameters at prearranged position ($j = 0$) according to sensitivity levels and using K-means clustering technique ($K = 3$).	60
7	Results of the DoEs for the rectangular flange normalized taking \mathcal{C}_1 as a reference. . .	62
8	Scores of DoEs result for the rectangular flange.	62
9	Overall framework for fitting a new CAD model using the best configuration of a similar CAD model found in the database.	63
10	Point cloud (b) obtained while scanning a real gearbox (a) without disassembling it and using a ROMER Absolute Arm 7520 SI laser scanner.	63
11	Four CAD models before and after the fitting with their best configuration resulting from the DoEs: (a_1) coarse pre-arrangement of the rectangular flange in the virtually-generated point cloud; (a_2) globally fitted rectangular flange; (b_1) pre-arrangement of the gland; (b_2) locally fitted gland in the virtually-generated point cloud; (c_1) pre-arrangement of the chain link; (c_2) locally fitted chain link in the virtually-generated point cloud of a chain assembly; (d_1) pre-arrangement of the round flange; (d_2) locally fitted round flange in the real scanned point cloud of a gearbox assembly.	64
12	Scores obtained from averaging the results obtained from DoEs performed 3 times for the selection of best configuration of: (a) rectangular flange; (b) gland; (c) chain link and (d) round flange.	65
13	Global fitting of bend joint defined by 8 control parameters: (a) initial configuration after the prearrangement step, (b) final fitted part minimizing the deviation with the point cloud and using best configuration \mathcal{C}_6	65
14	Similarity check between the query model and the models in database.	66
15	Resulting scores for the 18 configurations used for the SA-based fitting of bend joint. .	66

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

LIST OF FIGURES

1	Framework for the new reverse engineering technique for efficient reconstruction of the mechanical parts/assembly models.	70
1	Overall framework for the new reverse engineering technique exploiting a generic multi-dimensional fitting module.	75
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^κ in the vicinity of the reference profile PR_1^κ ; (c) final 2D parametric sketch \mathcal{S}_1^κ fitted in the filtered profile $PR_1^{\kappa,2}$; (d) draft CAD model \mathcal{M}_1^κ extruded from the fitted 2D parametric sketch \mathcal{S}_1^κ with an estimated extrusion length; (e) 3D CAD model \mathcal{M}_1^κ fitted in the filtered point cloud $PC_1^{\kappa,2}$; (f) fitted assembly \mathcal{A}_1^κ containing two models \mathcal{M}_1^κ and \mathcal{M}_2^κ assembled together with coaxiality and contact constraints.	76
3	Generic multi-dimensional fitting process handling parameters and constraints at the level of sketches, parts and assemblies.	77
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).	77
5	Definition of the parameters controlling the gland of a pump: 5 parameters for the 2D sketch and 3 parameters for the 3D features.	77
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 $PR_i^\kappa(t)$ with a protrusion not to be considered during the fitting; (b) points of $\mathcal{S}_i^\kappa(t)$ and $PR_i^\kappa(t)$ 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 $PR_i^{\kappa,2}(t)$ defining the point cloud used for the computation of the energy.	78
7	Successive steps of the reverse engineering of a Wankel engine with a point cloud generated from an as-scanned virtual scanning.	80

LIST OF FIGURES

8	Simulated annealing evolution curves for the three parameters groups to reconstruct the parametric sketch of the casing part.	81
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.	83
A Annex		
1	User interface of prototype plugin for SolidWorks.	96
2	Selection of optimization strategy.	97
3	Selection of optimization strategy.	97
4	Selection of optimization algorithm either with ICP or without it.	98
5	Setting default values for the optimization.	98
6	Selecting free parameter's values.	99
7	Options selection for the results section.	99
8	Accessing plugin form SolidWorks.	100
9	Opening the CAD model and point cloud in the CAD modeler.	101
10	Setting optimization algorithm with default parameters and selection of the free parameters from CAD model.	102
11	Linking parameters of CAD model to Simulated Annealing to optimize during the optimization process.	103
12	Fitted Sonotrode after the optimization.	103

Abbreviation

2D Two Dimensional

3D Three Dimensional

B-Rep Boundary Representation

CAD Computer-Aided Design

CB-CTT Curriculum-based course timetabling

CSG Constructive Solid Geometry

DMU Digital Mock-Up

DoEs Design of Experiments

HPR Hidden Point Removal

HSDM Hierarchical Space Decomposition Model

ICP Iterative Closest Point

ICT Iterative Closest Triangle-point

LBSA List-based Simulated Annealing algorithm

STL Standard Tessellation Language

PC Point Cloud

PDP Product Development Process

PSO Particle Swarm Optimization

RANSAC Random Sampling Consensus

RE Reverse Engineering

SA Simulated Annealing

TSP Traveling Salesman Problem

MC Maximum Clique

ABBREVIATION

Résumé

Même si de nombreuses techniques de rétro-ingénierie existent pour reconstruire des objets réels en 3D, très peu sont capables de traiter directement et efficacement la reconstruction de modèles CAO éditables d'assemblages de pièces mécaniques pouvant être utilisés dans les étapes du processus de développement de produits (PDP). En l'absence d'outils de segmentation adaptés, ces approches ont des difficultés à identifier dans le modèle reconstruit les différentes pièces qui composent l'assemblage. Cette thèse vise à développer une nouvelle technique de rétro-ingénierie pour la reconstruction de modèles CAO modifiables. L'originalité réside dans l'utilisation d'un processus de calage basé sur l'algorithme d'optimisation de recuit simulé et exploitant un filtrage à deux niveaux capable de capturer et de gérer les limites des géométries des pièces à l'intérieur du nuage de points global. Cette approche permet ainsi la détection des interfaces et l'ajustement local d'un modèle de pièce au nuage de points. La méthode proposée exploite différents types de données (par exemple, des nuages de points, des modèles CAO éventuellement stockés dans une base de données avec les meilleures configurations de paramètres associées pour le processus calage). L'approche est modulaire et intègre une analyse de sensibilité pour caractériser l'impact des variations des paramètres d'un modèle CAO sur l'évolution de la déviation entre le modèle CAO lui-même et le nuage de points à caler. L'évaluation de l'approche proposée est réalisée en utilisant à la fois des nuages de points scannés réels et des nuages de points générés virtuellement, qui comportent plusieurs artefacts pouvant apparaître avec un scanner réel. Les résultats couvrent plusieurs scénarios d'application liés à l'industrie 4.0, allant de l'ajustement global d'une seule pièce à la mise à jour d'une maquette numérique complète intégrant des contraintes d'assemblage. L'approche proposée est particulièrement adaptée pour l'aider au maintien de la cohérence entre un produit/système et son jumeau numérique.

Mots-clés : rétro-ingénierie, calage 2D et 3D, recuit simulé, jumeau numérique, industrie 4.0, analyse de sensibilité, paramètres de modèles CAO, nuages de points scannés, segmentation.

RESUME

Abstract

Even if many Reverse Engineering techniques exist to reconstruct real objects in 3D, very few are able to deal directly and efficiently with the reconstruction of editable CAD models of assemblies of mechanical parts that can be used in the stages of Product Development Processes (PDP). In the absence of suitable segmentation tools, these approaches struggle to identify and reconstruct model the different parts that make up the assembly. The thesis aims to develop a new Reverse Engineering technique for the reconstruction of editable CAD models of mechanical parts' assemblies. The originality lies in the use of a Simulated Annealing-based fitting technique optimization process that leverages a two-level filtering able to capture and manage the boundaries of the parts' geometries inside the overall point cloud to allow for interface detection and local fitting of a part template to the point cloud. The proposed method uses various types of data (e.g. clouds of points, CAD models possibly stored in database together with the associated best parameter configurations for the fitting process). The approach is modular and integrates a sensitivity analysis to characterize the impact of the variations of the parameters of a CAD model on the evolution of the deviation between the CAD model itself and the point cloud to be fitted. The evaluation of the proposed approach is performed using both real scanned point clouds and as-scanned virtually generated point clouds which incorporate several artifacts that could appear with a real scanner. Results cover several Industry 4.0 related application scenarios, ranging from the global fitting of a single part to the update of a complete Digital Mock-Up embedding assembly constraints. The proposed approach presents good capacities to help maintaining the coherence between a product/system and its digital twin.

Keywords : reverse engineering, 2D and 3D fitting, Simulated Annealing, digital twin, industry 4.0, sensitivity analysis, CAD model parameters, as-scanned point clouds, segmentation.

ABSTRACT

Sommario

Nonostante l'esistenza di molte tecniche di Reverse Engineering per ricostruire modelli digitali 3D di oggetti reali, pochissimi metodi sono in grado di ottenere in modo diretto ed efficiente la ricostruzione di modelli CAD di assiemi di parti meccaniche modificabili e quindi pienamente utilizzabili nelle varie fasi del Processo di Sviluppo di Prodotto (PDP). In assenza di strumenti di segmentazione adeguati, questi approcci hanno difficoltà ad identificare e ricostruire il modello delle diverse parti che compongono l'assieme. Questa tesi mira a definire una nuova tecnica di Reverse Engineering per la ricostruzione di modelli CAD parametrici mediante l'uso di modelli di riferimento. L'originalità risiede nell'uso di un processo di ottimizzazione che sfrutta la tecnica di simulated annealing ed un filtraggio a due livelli in grado di catturare e gestire i confini delle singole geometrie all'interno della nuvola di punti completa per consentire il rilevamento dell'interfaccia e l'adattamento locale del modello di una singola parte alla nuvola di punti. Il metodo proposto è applicabile a vari tipi di geometrie, ovvero sezioni 2D, parti singole e assemblati ed utilizza vari tipi di dati (nuvole di punti, modelli CAD, parametri relativi alle geometrie da ricostruire e al processo di adattamento).

Il metodo proposto presenta quindi un approccio modulare in due fasi. La prima fase mira ad identificare le interfacce tra le parti sfruttando e combinando un insieme di caratteristiche all'interno dei dati disponibili (nuvole di punti e modelli CAD). La seconda fase gestisce l'adattamento dei modelli CAD di riferimento alle corrispondenti porzioni delle nuvole di punti scansionate. Il metodo integra l'analisi di sensitività per caratterizzare l'impatto delle variazioni nei parametri del modello CAD di riferimento sull'evoluzione della deviazione tra il modello CAD stesso e la corrispondente nuvola di punti. Al fine di facilitarne l'utilizzo, la tesi propone inoltre un approccio case-based per l'identificazione ed utilizzo delle configurazioni ottimali dei parametri in base alla forma dell'oggetto considerato.

L'approccio proposto è stato valutato utilizzando sia nuvole di punti ottenute da reali scansioni che nuvole di punti generate virtualmente (as-scanned) che incorporano diversi artefatti così come

SOMMARIO

potrebbero apparire nel caso di acquisizione con un vero scanner. I risultati sono di interesse per diversi scenari applicativi correlati all'Industria 4.0, che vanno dall'adattamento globale di una singola parte all'aggiornamento di un assieme completo con relativi vincoli tra le parti. Pertanto l'approccio proposto presenta caratteristiche idonee a supportare la coerenza tra un prodotto / sistema e il suo gemello digitale.

Parole chiave : reverse engineering, 2D and 3D fitting, Simulated Annealing, gemello digitale, industria 4.0, analisi di sensitività, modelli CAD parametrici, nuvole di punti, segmentazione.

Publications

Journal

- **As-scanned point clouds generation for virtual reverse engineering of cad assembly models**
Jérémy Montlahuc, Ghazanfar Ali Shah, Arnaud Polette and Jean-Philippe Pernot
Computer-Aided Design and Applications 2018, 16(6):1171–1182.
- **Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies**
Ghazanfar Ali Shah, Arnaud Polette, Jean-Philippe Pernot, Franca Giannini, Marina Monti
Engineering with Computers 2020:1–19.

Submitted in Journal

- **User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models**
Ghazanfar Ali Shah, Arnaud Polette, Jean-Philippe Pernot, Franca Giannini, Marina Monti
Submitted to journal *Advances in Engineering Software*, May 2021
- **Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications**
Ghazanfar Ali Shah, Arnaud Polette, Jean-Philippe Pernot, Franca Giannini, Marina Monti
Submitted to journal *Computers in Industry*, May 2021

Conferences

- **Sensitivity Analysis for Simulated Annealing-Based Fitting of CAD Models to Point Clouds of Digitized Parts (Best paper award in international conference)**
Ghazanfar Ali Shah, Arnaud Polette, Jean-Philippe Pernot, Franca Giannini, Marina Monti
In *Advances on Mechanics, Design Engineering and Manufacturing III: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2020*, June 2–4, 2020 2021 (p. 274). Springer Nature.
- **Fitting parametric 2D & 3D CAD models to point clouds of digitalized mechanical assemblies for reverse engineering application**
Ghazanfar Ali Shah, Jean-Philippe Pernot, Franca Giannini, Marina Monti, Arnaud Polette
The European Association for Computer Graphics 40th Annual Conference - EUROGRAPHICS 2019, pp. 11-12, Genova, Italy, 6 - 10 May, 2019.

— **Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies**

Ghazanfar Ali Shah, Jean-Philippe Pernot, Franca Giannini, Marina Monti, Arnaud Polette

GTMG 2020: Journées du Groupe de Travail en Modélisation Géométrique, Aix-En-Provence, France, 02 - 03 July, 2020.

PUBLICATIONS

Introduction

Reverse Engineering

In the past few years, there has been a lot of research in the field of Reverse Engineering (RE). RE is the process of turning the information of any physical part or assembly into the digital form with the help of dedicated hardware. It mainly involves two steps: data acquisition and data processing for 3D model creation. There exists several techniques being used (e.g. contact and non-contact) for the acquisition of point clouds. Different devices are available starting from movable portable laser arm scanner to a big fixed bridge type scanners. Each technology has its own advantages and disadvantages with different applications and usages.

These devices can accurately measure and collect data from objects in the form of points on its surface expressed by the coordinates x, y and z. The data is collected in a cloud with millions of points. The point cloud may vary in size depending on the object dimension and the quality of scanning. The process is illustrated in Figure A1 in which a car model is being scanned using a portable multi-axis laser scanner. The resulting point cloud is composed of about forty million points which are later transformed into a CAD model after the data treatment step.

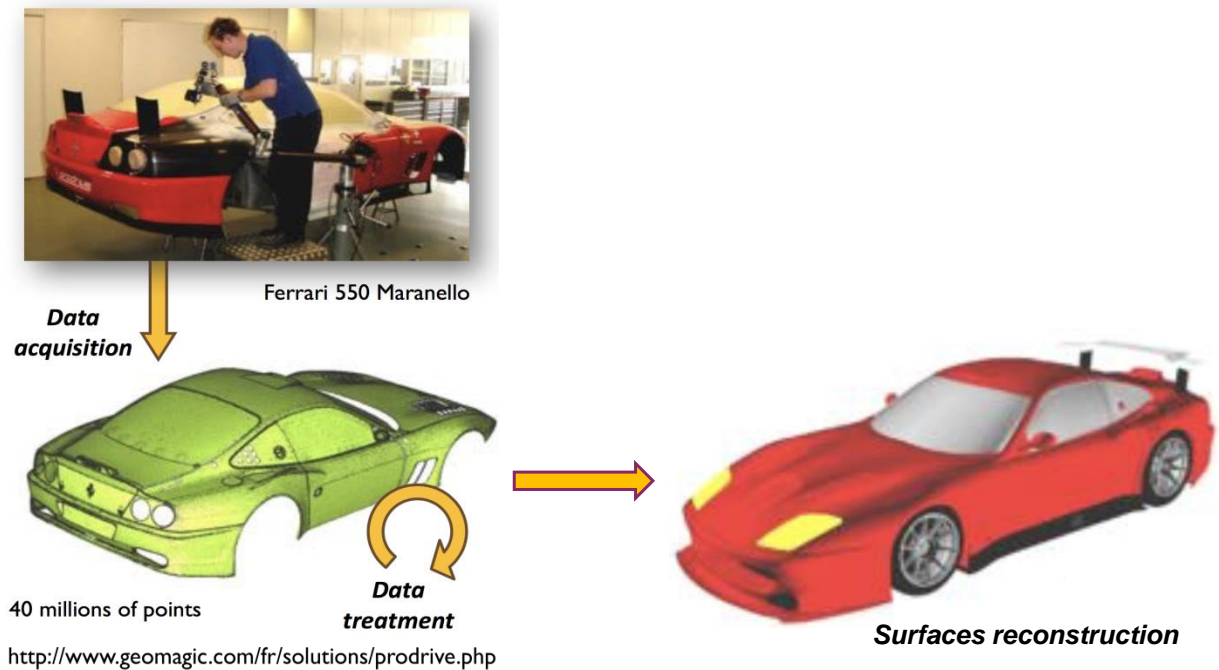


Figure A1 – From data acquisition of an object to the CAD model.

Reverse Engineering applications

Over the last two decades, Reverse Engineering usage has spread significantly and it has caused a radical change in the way the manufacturing of components is perceived today. In the industrial context, RE has gained a considerable momentum for a variety of applications and the integration of CAD tools has further amplified its dominance. Examples of RE engineering within the product development and usage life-cycles include the updating of digital models after physical prototyping, the manufacture of parts no longer in production, the repairing of existing products and evaluation solution alternatives and, more recently, the alignment of a digital twin to physical counterpart.

RE benefits have also opened new doors for exploring incomplete designs that do not have much information. To fill the voids in the design information, RE is applied widely to discover the potential design data that might not have been documented. For manufacturers, having accurate design data documentation ensures production efficiency in computer-aided manufacturing. RE is being used for product failure investigation and extending the product's functional life cycle period. It helps in the identification of premature failure to improve the product. Reverse Engineering in product development saves a lot of time and reduces cost [1].

Moreover, the concept of reverse engineering is not simply restricted to analyzing product designs. It is also focused on making a product easy and fast to manufacture without compromising on its performance and quality. The competition has raised as the manufacturers take advantage of reverse engineering to study the products manufactured by their competitors and to stand equal against the competitors, manufacturers are forced to bring innovations in their product.

Traditional Reverse Engineering of CAD models

There are many commercial software such as CATIA, RapidForm, Geomagic that support traditional RE methodology to transform the raw point cloud data to a CAD model. Figure A2 shows the process of traditional reverse engineering that consists of the following steps: data acquisition, pre-processing (noise filtering and merging), triangulation, segmentation, and surface fitting to obtain CAD models [2]. Once the existing objects are digitized, the resulting point cloud also incorporates noise. The level of noise depends on the quality of the scanner and the way objects are digitized. The raw point cloud requires pre-processing to filter the noise. Merging operation can also be required if multiple data sets are belonging to a single object. Next, point clouds are segmented to extract surfaces in a patch-by-patch

manner. Later these extracted patches of surfaces are trimmed and assembled to get the final CAD model.

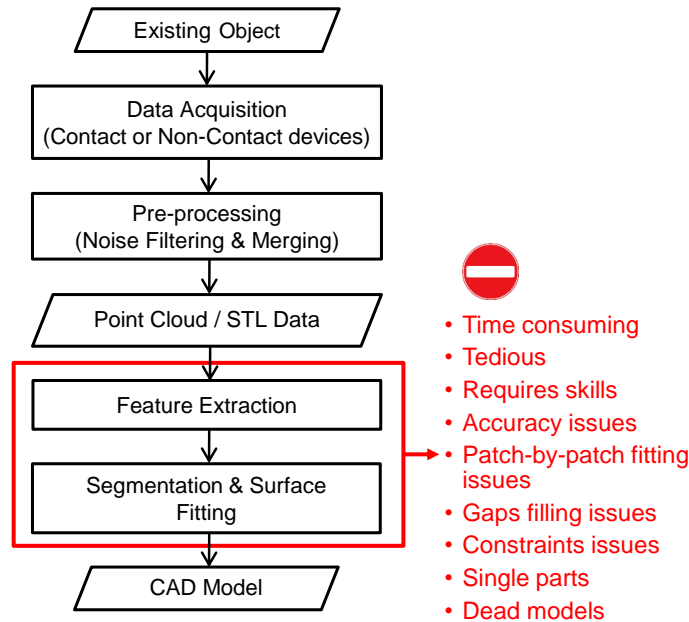


Figure A2 – Workflow for the traditional reverse engineering process, and possible difficulties.

Industrial problem

In the existing reverse engineering techniques, it is very difficult to reconstruct editable CAD parts or assemblies that can later be used and modified in the Product Development Process (PDP). Following the traditional RE techniques engineers have to face many issues: pre-processing of the data, segmentation of point clouds, decomposition in patches, fitting of primitives, trimming and connection of the resulting surfaces. Especially, the two steps related to *Feature Extraction* and *Segmentation & Surface Fitting* (highlighted in red in Fig. A2) are difficult to handle. The process is cumbersome and tedious and it requires really good skill to handle the patch-by-patch and accuracy issues. More often, users have to face issues related to leftover gaps while joining the surface patches and to maintain the constraints amongst different features while reconstructing. Most of the work is related to the reconstruction of single parts and few people are working on the reconstruction of the assembly models. At the end of this sequential time-consuming patch-by-patch reconstruction strategy, designers generally produce “dead” models that cannot be later modified as needed. This is an important limitation of the existing techniques which prevents efficient reconstruction of consistent and editable CAD models.

Thesis Objective

To overcome the above limitations, this thesis aims at proposing a new reverse engineering technique allowing the creation of editable CAD models from the point cloud of scanned mechanical parts and assemblies by starting either from scratch using a rough sketch or by using reference parametrized models.

Research sub-problems

To achieve the above objective, the research question has been decomposed in sub-problems to be solved.

Collection of testing cases and methods for evaluating the studied methods)

- In order to verify the correctness of the reconstructed parametric parts and assemblies from a given point cloud, the correct and precise values of the parameters need to be known. This is not possible with point clouds obtained from real laser scanner acquisition. Therefore, methods simulating the scanning process over known parametrised CAD models have to be defined. This process is referred as Virtual scanning. It involves the specification of the necessary processing steps for the generation of as-scanned point clouds incorporating artifacts. This will save an immense amount of time for scanning real objects with the laser scanners, and it will enable the validation of the new reverse engineering technique while comparing the values of the parameters of the reverse-engineered part to their values in the CAD model that serve as a reference to generate the so-called virtual point cloud.

Fitting technique

- Specification of the necessary processing steps and a capable workflow for the CAD model adaptation and fitting to the point cloud. The idea is to have an optimization algorithm able to change the dimensions of the CAD models so as to fit to a point cloud and minimize the distance between the two;
- Specification of the optimization method and tool for the fitting process;
- Identification of the default setting parameters for the selected optimization method to obtain the best fitted geometries.

Segmentation technique

- Development of a segmentation technique to allow the identification of the boundaries between parts and the point cloud portion corresponding to a part within the assembly in order to allow local fitting.

User-Driven Computer-Assisted Reverse Engineering (New Reverse Engineering technique)

- Identification and combination of the geometric characteristics (e.g. geometric constraints, CAD parameters, features type) with information available in the reference point clouds (e.g. information of the coordinates of points and their normal vector) for a User-Driven Computer-Assisted Reverse Engineering of CAD models.

Methodology

To achieve the objective of the thesis, the proposed methodology for the specification of new reverse engineering tools and methods is illustrated in Figure A3 and it has been broken down into the following parts reflecting the above research challenges:

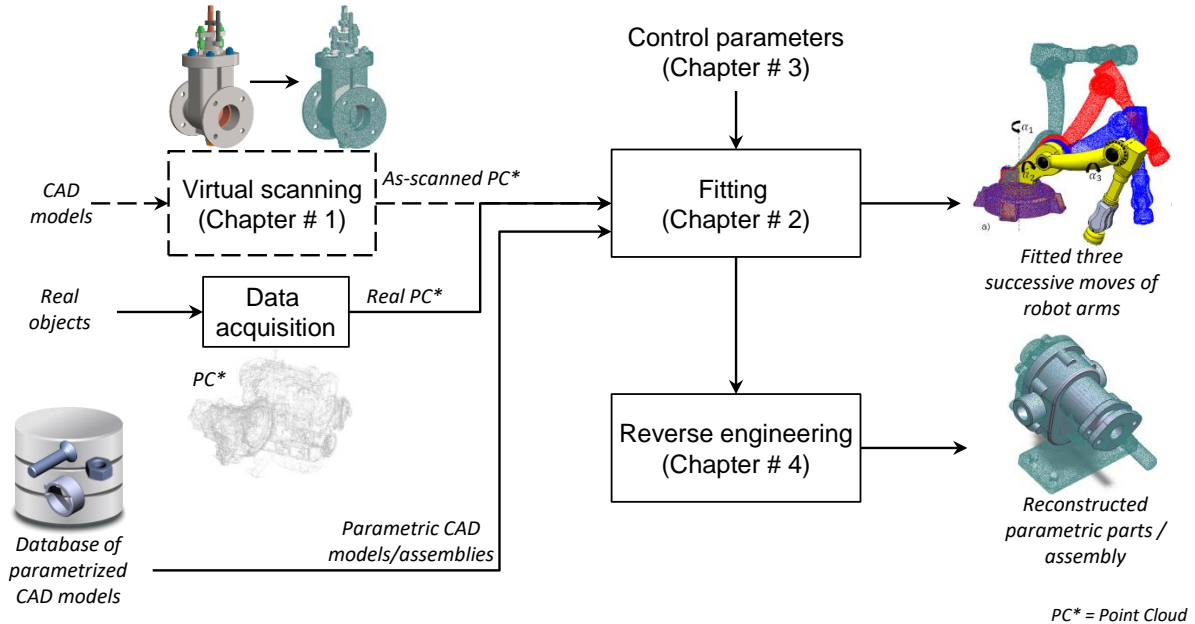


Figure A3 – Overall framework for the RE to reconstruct mechanical parts/assembly models.

- Creation of a technique to generate point clouds that will be further used for the validation of proposed solution. To reverse engineer many examples for the reconstruction of single and

multiple parts, a database of point clouds is required. For this purpose, there is a need to develop an as-scanned point cloud generation technique to create a large number of point clouds that also incorporate different artifacts as if they are obtained from real scanners. Such a technique will save a great amount of time by generating virtual point clouds of parts and assemblies in few minutes. The database will also include real point clouds obtained using a laser scanning device;

- A segmentation technique required to allow local fitting of parts and sub-assemblies by segmenting the point cloud of the assembly model during the reconstruction process;
- Reconstruction and fitting of CAD models. The idea here is to make use of the existing parametric CAD models whose parameters are to be adapted to fully fit into a given point cloud. The fitting process is governed by a numerical optimization method that modifies the values of the parameters of the CAD models in order to minimize with the corresponding point cloud elements;
- Enhancement of the optimization-based fitting. Here methods have to be defined for the selection of the parameters of the adopted optimization algorithm to directly get good fitting results. This would help users to set and select default parameters of the optimization algorithm that would produce the best fitting results for new part to be reconstructed;
- The new reverse engineering framework to reconstruct editable CAD models and assemblies. The idea here is to make the best use of the tools and knowledge acquired in the previous stages for the new template-based reverse engineering technique. The method should be able to reconstruct parametric 2D sketches, 3D parts and assemblies.

Scientific contributions

To tackle the identified research problems, this thesis has been divided into four chapters and each chapter is based on articles. A brief description of the scientific contributions and their link to the corresponding chapter is listed in the following Table 1.

Scientific contributions	Corresponding chapters
1. Virtual scanning (i) New technique for the generation of as-scanned point clouds that incorporates artifacts as if they are generated from real scanning process.	Chapter 1 As-scanned point clouds generation for virtual reverse engineering of cad assembly models.
2. Fitting (i) A new part-by-part fitting framework allows the fitting of parameterized and editable CAD models to point clouds of digitized mechanical parts assemblies; (ii) Both local and global fitting can be performed thanks to a segmentation strategy controlled by a simulated annealing; (iii) The method can fit single parts as well as multiple parts optionally constrained together with assembly constraints and fitted simultaneously.	Chapter 2 Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies.
3. Enhancement of fitting algorithm (i) A new framework using design of experiments (DoEs) for the creation of a database of CAD models stored together with their best SA-based fitting configurations; (ii) Sensitivity analysis to study the influence of SA on the parameters of CAD models during the fitting process; (iii) Identification of the default setting parameters for SA-based fitting to obtained best-fitted geometries using DoEs.	Chapter 3 Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications.
4. New Reverse Engineering (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 constraints; (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 interfaces detection.	Chapter 4 User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models.

Table 1 – Connections of the four chapters with the scientific contributions.

Manuscript organization

The proposition of this thesis and how it has been achieved is described in chapters 1-4 as outlined in Table 1:

- **Chapter 1** details the as-scanned point cloud generation technique in which point clouds are generated from the existing CAD models and assemblies incorporating various artifacts that would appear if they were scanned with real scanners. Generation of such point clouds helps in the investigation and validation of New Reverse Engineering technique for the reconstruction of CAD parts and assemblies from their corresponding PC.
- **Chapter 2** introduces a new fitting approach for efficient part-by-part reconstruction of editable CAD models fitting in the point clouds obtained from both the real scanner and as-scanned point cloud generation technique (Chapter 1). The proposed fitting process relies on the Simulated Annealing algorithm that minimizes the deviations between the CAD models and the point cloud. The method is validated on the local and global fitting of CAD models including single parts and assemblies.
- **Chapter 3** explains the third article on the enhancement of the Simulated Annealing-based fitting strategy by identifying its optimal default setting parameters for the fitting of geometries in the reference point clouds. The proposed method uses sensitivity analysis to study the impact of Simulated Annealing on the parameters of the CAD models during the optimization process. To find the optimal default setting parameters of Simulated Annealing-based fitting, the DoEs approach is used to identify and store the best configuration of parameters that would produce the best fitting result and can also be used in the future for the reconstruction of new parts using a similarity assessment tool.
- **Chapter 4** introduces a novel reverse engineering technique that allows the reconstruction of editable CAD models from a given point cloud. Depending on the different scenarios on the available data sets and starting point and in addition to what is shown in Chapter 2, the new reverse engineering also handles the fitting of 2D parametric sketches that can later be used for the reconstruction of 3D features. Differently from Chapter 2 in which the segmentation is driven by a distance criterion, this chapter also details a two-level filtering technique for the

segmentation to capture and manage the boundaries of the geometries inside the overall point cloud in order to allow for local fitting and interface detection.

Current limits and possible evolutions are discussed in the **Conclusions and perspectives** section. Finally, **Annex A** provide details on the software prototype developed as a plugin of SolidWorks to ease the application of the proposed RE technique.

CHAPTER

1

As-scanned point clouds generation for virtual Reverse Engineering of CAD assembly models

Contents

1.1 Introduction	12
1.2 Conclusion	27

1.1 Introduction

To test and enhance the efficiency of the new reverse engineering method (Chapters 2 and 4), there was a need to have several point clouds related to single parts and assemblies. To use real laser scanners usually takes a great amount of time to generate point clouds of the real objects. This chapter introduces a new method of as-scanned point cloud generation technique from the available datasets of CAD models and assemblies (highlighted in green in Fig. 1).

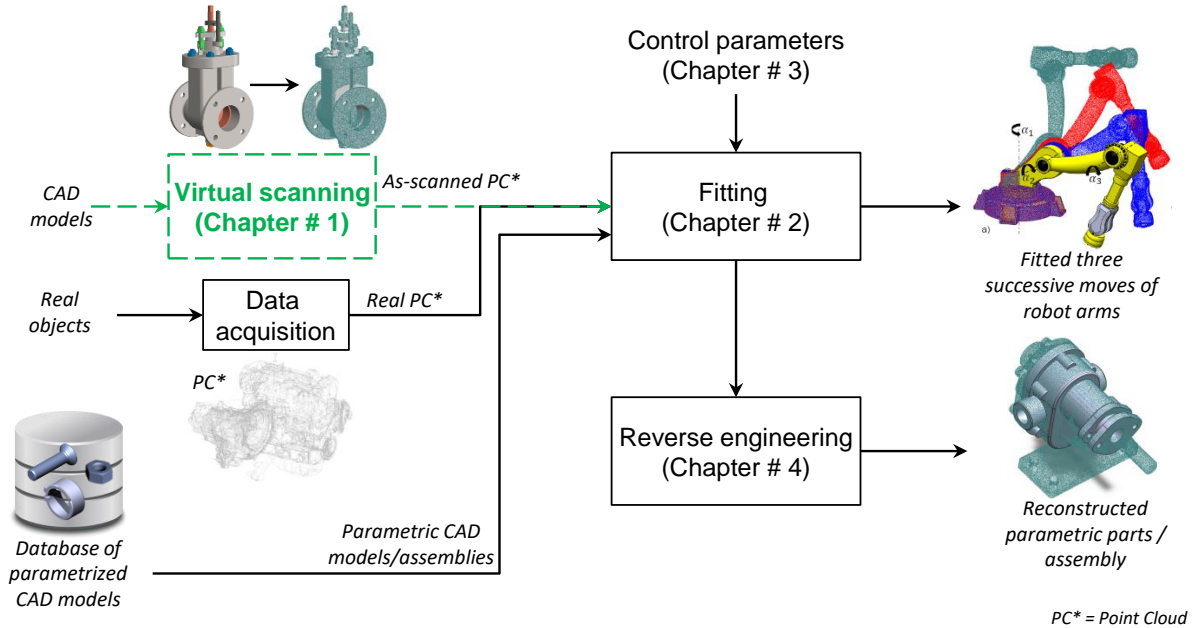


Figure 1 – As-scanned point cloud generation supporting the fitting and new RE technique to reconstruct mechanical parts/assembly models.

We can access a large number of models from the different available data sources like GrabCAD [3], TraceParts [4], and 3DmodelSpace [5], etc. The generated point clouds are incorporated with different artifacts (e.g. non-uniform sampling, missing data, misaligned point clouds, noisy data, and outliers) to be more realistic and as if the obtained point clouds are coming from real scanners. In few seconds, a large DMU can be transformed to a point cloud without using a real scanner. Moreover, the dimensions (parameter values) of the CAD models used for the generation of virtual point clouds are known before converting them into point clouds. This is interesting to compare those parameter values to the ones obtained through the fitting process, and thus to validate the approach. The proposed tool for the generation of as-scanned point cloud is controlled by several input parameters that can be tuned to

1.1. INTRODUCTION

match the output result with available data acquisition devices. Such a tool helps in the generation of a small database of point clouds that can serve many requirements.

1.1. INTRODUCTION



As-scanned Point Clouds Generation for Virtual Reverse Engineering of CAD Assembly Models

Jérémy Montlahuc¹ , Ghazanfar Ali Shah¹ , Arnaud Polette¹  and Jean-Philippe Pernot¹ 

¹Arts et Métiers, LISPEN, HeSam, Aix-en-Provence, France

Corresponding author: Jean-Philippe Pernot, jean-philippe.pernot@ensam.eu

Abstract. This paper introduces a new approach for the generation of as-scanned point clouds of CAD assembly models. The resulting point clouds incorporate various realistic artifacts that would appear if the corresponding real objects were digitalized with a laser scanner. Such a virtual Reverse Engineering technique can produce a huge amount of realistic point clouds much faster than using classical time-consuming Reverse Engineering techniques on real physical objects. Here, there is no need to use a laser scanner and the post-processing steps are automatic. Using this technique, it is easy to create large databases of point clouds automatically segmented and labeled from the CAD models and which can be used for supervised machine learning. The proposed approach starts by generating a triangle mesh wrapping the CAD assembly model to be reverse engineered. The resulting watertight mesh is then sampled to obtain a more realistic distribution of points. The occlusion phenomenon is then simulated using a hidden point removal algorithm launched from several viewpoints. A misalignment procedure can optionally be used to simulate the fact that in real-life Reverse Engineering the position and orientation of the laser scanner and/or real object would have been changed to get a different scanning viewpoint. The virtual Reverse Engineering process ends by generating noise and by inserting outliers. The approach is illustrated and validated on several industrial examples.

Keywords: as-scanned point clouds, scanning artifacts, surface wrapping, sampling, hidden point removal, noise and outliers, reverse engineering.

DOI: <https://doi.org/10.14733/cadaps.2019.1171-1182>

1 INTRODUCTION

Over the last years, the use of artificial intelligence techniques to analyze and process geometric models has become a new trend in computer sciences. This is notably true when considering the segmentation and classification of 3D point clouds [5][9]. However, such techniques require the access to large datasets which may also have to be labeled when considering supervised learning

techniques. Therefore, one of the key issues when defining such learning approach is to be able to rely on available and trustable large datasets. This is not straightforward as it can take a lot of time to generate and process all the data. Actually, in Reverse Engineering process [8] the acquisition and processing time can range from a few minutes to several hours [1]. This strongly depends on the adopted technology (e.g. LIDAR, laser scanner, structured-light scanner, RGB-D sensor), the acquisition procedure followed by the operator and the complexity of the object to be reverse engineered. For example, scanning a simple sonotrode from multiple viewpoints (Fig. 1.a1) and treating (e.g. cleaning, filtering, registration, simplification, meshing) the resulting point clouds (Fig. 1.a2) can require up to several tens of minutes. Thus, it becomes unreasonable to try to manually generate thousands of point clouds using classical Reverse Engineering techniques on more complex existing physical objects or environments. Moreover, to be able to analyze and understand the impact of both the type of sensors and adopted control parameters, it is also important to have access to multiple point clouds of the same object following various acquisition scenarios. This further explodes the number of required acquisitions and associated treatments, thus justifying the need to develop the fully virtual Reverse Engineering technique presented in this paper.

In the proposed virtual Reverse Engineering approach, point clouds are automatically generated from CAD models of parts or assemblies. Thus, our approach can make use of available databases containing a huge amount of CAD models (e.g. GrabCAD, TraceParts, 3DModelSpace). The resulting point clouds incorporate various realistic artifacts that would appear if the corresponding real objects were digitalized with a real acquisition device. If the CAD models are labeled or enriched with semantic information, the generated point clouds could easily inherit from the available information. The inheritance procedure is not developed in this paper which focuses on the way point clouds can be realistically generated. As the approach is fully parameterized, for a given CAD model, several point clouds (Fig. 1.b1 to 1.b4) can be generated to simulate different scanning conditions (e.g. type of acquisition device and associated control parameters, environmental conditions, adopted acquisition sequence). As a consequence, a large variety of as-scanned point clouds can be generated in few second.



Figure 1: Real sonotrode (a1) scanned to get point clouds that have been post-treated (a2). CAD model of a sonotrode (b1) virtually reverse engineered following three parameterizations of the proposed framework (b2-b4).

To generate realistic as-scanned point clouds, it is important to analyze and to understand the multiple artifacts which can appear during a real reverse engineering session. Actually, artifacts can result from more or less complex phenomena generated by, and/or between, the acquisition device (e.g. type, control parameters), the digitalized object (e.g. material, color, shape, size), the operator (e.g. acquisition strategy, experience), and the environment (e.g. light, temperature, vibration). However, even

if the origins can be multiple, the impacts on the resulting point clouds can be classified according to five main categories (Fig. 2): non-uniform sampling, missing data, misaligned point clouds, noisy data and outliers [2].

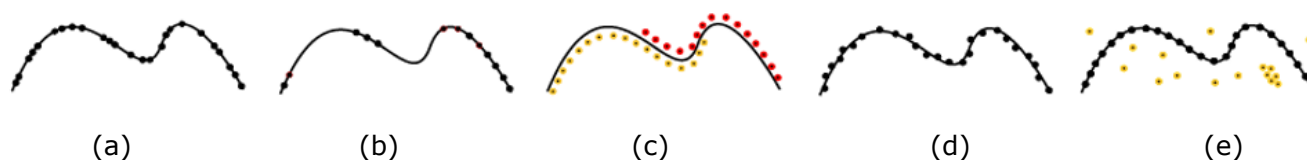


Figure 2: Different artifacts: (a) non-uniform sampling, (b) missing data, (c) misaligned scans, (d) noisy data and (e) outliers[2].

This paper introduces a modular framework for the generation of as-scanned point clouds incorporating the above mentioned artifacts. Section 2 introduces the overall framework and discusses some of its characteristics. The modules and associated control parameters are detailed in section 3. Section 4 presents the tested configurations, the adopted evaluation criteria as well as the results on several industrial examples. The last section concludes this paper and introduces several perspectives.

2 OVERALL FRAMEWORK

The proposed approach is composed of several modules (Fig. 3). It starts by generating a triangle mesh wrapping the CAD model to be reverse engineered. The CAD model can either be a single part or an assembly of several parts. The resulting watertight mesh is then sampled to obtain a more realistic distribution of points. The occlusion phenomenon is then simulated using a hidden point removal algorithm launched from several viewpoints. As a result, the resulting point cloud is incomplete and some data are missing. Then, a misalignment procedure can optionally be used to take into account the fact that in real-life reverse engineering the object is acquired from several viewpoints. Thus, this procedure can modify the position and orientation of the point clouds' reference frames with respect to the reference frame of the original CAD model. The virtual Reverse Engineering process ends by generating noise and by inserting outliers.

Table 1 characterizes the different modules with respect to their ability to incorporate the above-mentioned artifacts. It clearly shows the complementarity of the different modules to achieve the desired results, i.e. being able to generate as-scanned point clouds incorporating artifacts appearing when scanning real-life physical objects. The approach is modular and each module is controlled by parameters which are detailed in the next section.

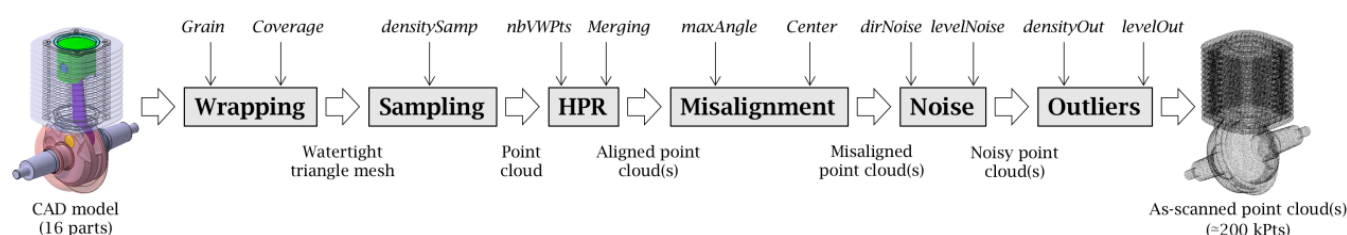


Figure 3: Virtual Reverse Engineering modular framework generating as-scanned point clouds from CAD assembly models.

		Modules					
		Wrap- ping	Sam- pling	HPR	Misalign- ment	Noise	Outli- ers
Artifacts	Non-uniform sam- pling	Yes	Yes	Yes	-	-	-
	Missing data	Yes	Yes	Yes	-	-	-
	Misaligned scans	-	-	-	Yes	-	-
	Noisy data	-	-	-	-	Yes	-
	Outliers	-	-	-	-	-	Yes

Table 1: Characteristics of the modules with respect to their ability to incorporate artifacts appearing when scanning real-life objects.

3 MODULES AND CONTROL PARAMETERS

This section briefly introduces the principles underpinning the proposed modules as well as the different control parameters which can be used to reflect as much as possible real scanning conditions.

3.1 Wrapping

The initial CAD model is wrapped to produce a watertight triangle mesh. As a result, some internal parts are not captured and some details of the resulting envelop can be simplified. This depends on the *Grain* accuracy and *wrapCoverage* ratio. The *Grain* characterizes the average distance between connected points of the resulting mesh. It can be set up according to the accuracy of the acquisition device to be simulated, for instance 50 μ m. The *wrapCoverage* determines the wrapping representation. A lower ratio will result in a thinner wrapping coverage. In the current implementation, the wrapping module makes use of CATIA V5 by Dassault Systèmes [4].

3.2 Sampling

The watertight triangle mesh resulting from the wrapping is then sampled to get a more realistic distribution of points. The amount of points is defined by the *densitySamp* expressing the number of points per square units. It can be set up according to the accuracy of the acquisition device to be simulated. The current implementation makes use of CloudCompare [3] to perform the sampling. Fig 4 illustrates the impact of the *densitySamp* parameter.



Figure 4: Sampling module: (a) watertight triangle mesh, (b) *densitySamp* = 2.5, (c) *densitySamp* = 5.

3.3 Hidden Point Removal (HPR)

To simulate multiple acquisition viewpoints, a simple and fast HPR operator can be run from several viewpoints. Following the approach of Khalifaoui et al. [7], a set of $nbVWpts$ viewpoints are selected from a predefined list of positions around the object to be virtually reverse engineered. The removal of hidden points is performed using the approach of Katz et al. [6]. The adopted HPR operator determines the visible points of the point cloud, as viewed from a given viewpoint (Fig. 5.b and 5.c). An optimized version of this algorithm has been used to speed up the identification of hidden points using an octree-based decomposition. Through the *Merging* output option, the user can decide whether he/she wants to get multiple incomplete point clouds (*Merging* = 0) or if the point clouds are merged to get a single output point cloud (*Merging* = 1). However, this module does not try to filter overlapping areas which may result from multiple viewpoints. This module is optional.

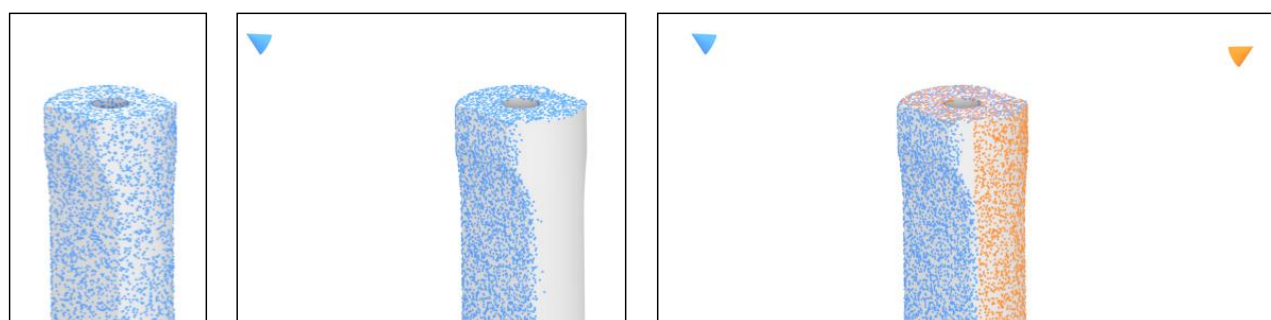


Figure 5: Hidden Point Removal module: (a) initial point cloud, (b) visible points after the HPR operator running from a given viewpoint shown with a cone, (c) from two viewpoints without merge (*Merging* = 0).

3.4 Misalignment

Since the HPR operator simply flags the visible points, the resulting point clouds perfectly fit in each other and there is no need to run an ICP algorithm. This differs from a real scan for which the point clouds acquired from different viewpoints appear in different reference frames. Thus, to simulate the fact that point clouds acquired with a real scanner would never fit perfectly, this module slightly rotates the point clouds one after the others. Of course, it can only be used if several (at least two) point clouds have been generated by the HPR module. Actually, a slight rotation is performed between two point clouds PC_i and PC_{i+1} generated from two successive viewpoints VP_i and VP_{i+1} . More precisely, PC_{i+1} rotates of an angle α_{i+1} that is defined by randomly selecting a value smaller than a user-specified *maxAngle*. As a default, the axis of rotation goes through the barycenter of the reverse engineered CAD model, and its direction is given by the plan's normal, defined by three points: the barycenter, the two viewpoints VP_i and VP_{i+1} . This misalignment procedure is run ($nbVWpts - 1$) times starting with the rotation of PC_2 . The barycenter used to define the axis of rotation can optionally be substituted by another user-specified *Center*. This can be of interest when the CAD model has widely varying main dimensions. Fig 6 illustrates the misalignment with three viewpoints.

3.5 Noise

Being able to insert noise is an important feature of the proposed virtual Reverse Engineering framework [10]. Noise depends on many factors (e.g. type of acquisition device, material of the object, orientation of the sensor with respect to the surface) but it can be characterized by three main parameters: the type of distribution law, the amplitude of the noise, and the direction of the noise. In this work, the noise distribution law is a uniform one.

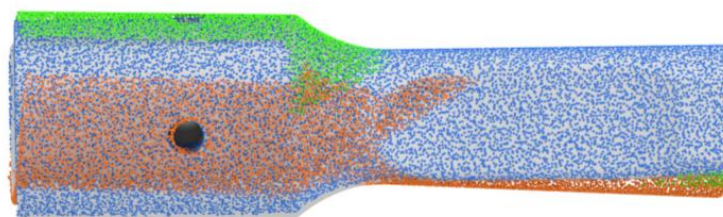


Figure 6: Misalignment module with three viewpoints, the barycenter as a rotation center and with the $maxAngle = 3.5$.

The remaining control parameters $levelNoise$ and $dirNoise$ (either along the normal to the surface, or along the line of sight) describe the way points are moved. The amplitude of the noise introduced at a given point is equal to $2 \times levelNoise \times (rand - 0.5)$ where $rand$ returns a single uniformly distributed random number in the interval $[0, 1]$. Fig 7 illustrates the impact of the $levelNoise$ parameter.

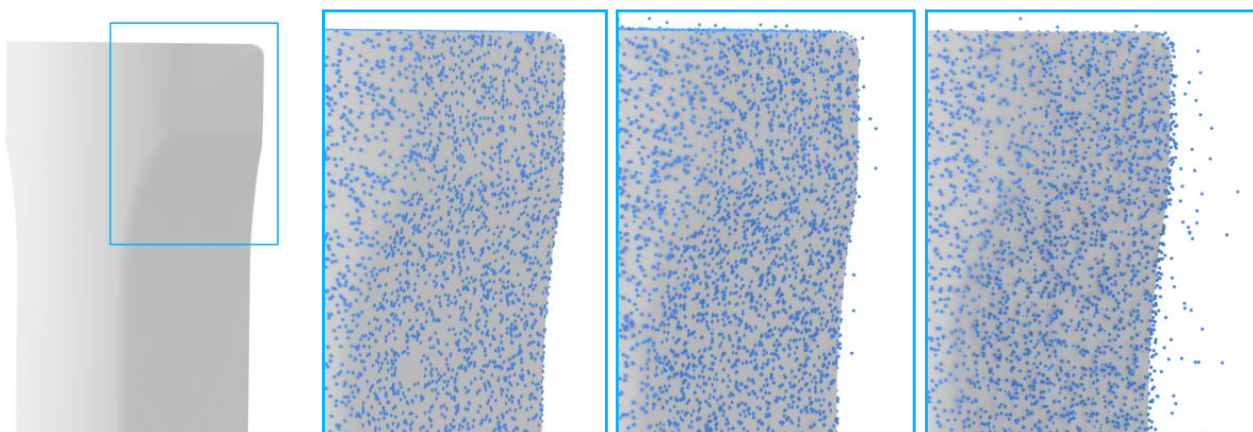


Figure 7: Noise module moving points along the normal to the surface ($dirNoise$): (a) zoom window, (b) $levelNoise = 0$, (c) $levelNoise = 0.03$, (d) $levelNoise = 0.1$

3.6 Outliers

Outliers are commonly due to structural artifacts in the acquisition process. In some instances, outliers are randomly distributed in the volume, where their density is smaller than the density of the points that sample the surface. Outliers can also be more structured, however, where high density clusters of points exist far from the surface. This can occur in multi-view stereo acquisition, where view-dependent specularities can result in false correspondences [2]. In the proposed approach, the idea is to identify a restricted set of points which can be duplicated and then moved randomly along the three directions of the space. Thus, the positioning of the outliers is driven by two control parameters: $densityOut$ and $levelOut$. The first one is a percentage of points to be duplicated and moved. Points are uniformly selected among the points of the point cloud(s). Outliers then result from a duplication of those selected points which are then moved in the three dimensions of the space using three distinct amplitudes each of them being computed using the formula: $20 \times levelOut \times (rand - 0.5)$.

4 RESULTS

This section presents several tests which have been performed to validate the proposed approach on several CAD assembly models to be virtually reverse engineered.

4.1 Tested configurations

To analyze the influence of the previously introduced control parameters and to show the potential of the proposed approach, four configurations have been tested. The parameters associated to those configurations are summarized in Table 2. Each configuration can be considered as a setting characterizing a scanning session (e.g. type of acquisition device, material of the object, lightning conditions, expertise of the operator) to be virtually simulated using the proposed approach. The links between the characteristics of a real scanning session and those parameters are not detailed in this paper. Due to space limitation, all the parameters are not tested.

Steps	Parameters	Configurations			
		1	2	3	4
Wrapping	Grain	0.5mm	0.5mm	0.5mm	0.5mm
	wrapCoverage	0	0	0	0
Sampling	densitySamp	5	5	8	1M pts
HPR	nbVWPts	1	4	2	6
	Merging	1	1	1	1
Misalignment	maxAngle	0°	0°	0°	0°
	Center	Default	Default	Default	Default
Noise	dirNoise	Normal	Normal	Normal	Normal
	levelNoise	0	0.03	0.1	0.02
Outliers	densityOut	0.1%	0.1%	0.1%	0.1%
	levelOut	0	0.03	0.1	0.02

Table 2: Parameters of four configurations characterizing different scanning sessions to be simulated using the proposed virtual Reverse Engineering framework.

4.2 Evaluation criteria and results

The point clouds resulting from the newly developed virtual Reverse Engineering framework are evaluated according to three criteria: the number of generated points, the deviation to the original CAD model (mean and max), and the coverage ratio. Actually, the coverage ratio represents the amount of available information when compared to what is available in the original CAD model. In our implementation, it corresponds to the ratio between the area of the final triangle mesh generated from the point cloud and the overall area of the initial CAD model. The coverage does not depend on the density of points but on the number of viewpoints which are to be used. Coverage has been computed with respect to the area of the entire CAD models (including hidden part) and the coverage with respect to the wrapping results (which could in this case reach 100%). The first coverage factor can be used also to analyze how complex the CAD models are, i.e. if they do have a lot of internal and hidden parts. The second coverage factor is used to analyze if a sufficient number of viewpoints has been used.

Following the configurations of Table 2, the virtual Reverse Engineering technique is applied on three first CAD models. Results are presented in Table 3 and in Figures 1, 8, 9. Points are represented by spheres. To be able to visualize the points in a proper way, i.e. without having fully overlapping spheres, the number of points has been reduced in each figure.

Models	Parts (#)	Config.	Points (#)	Deviation (mm)	Coverage wrt wrapping (%)	Coverage wrt assembly (%)
Sonotrode	1	1	18 936	0.000 (0.0)	30.8	30.8
		2	61 327	0.033 (2.2)	76.9	76.9
		3	44 631	0.127 (7.5)	53.8	53.8

Compressor	16	1	54 831	0.00012 (1.35)	67.4	38.8
		2	91 950	0.00302 (2.36)	14.6	8.4
		3	163 241	0.07916 (8.48)	17.4	10.0
Engine	82	1	106 475	0.00023 (1.14)	24.1	11.7
		2	109 173	0.01145 (3.85)	29.1	14.2
		3	158 042	0.10563 (8.27)	32.9	16.0

Table 3: Results of the virtual Reverse Engineering process applied on three first CAD models (sonotrode, compressor, engine) and following three different scanning configurations.



Figure 8: CAD model of a compressor reversed engineered with configurations 1 to 3.

To further validate the efficiency and robustness, the proposed approach has been tested on additional CAD parts and CAD assemblies as shown in Fig 10. Again, points are represented by spheres, and the number of points has been reduced in each figure to improve the visualization. To keep the consistency in the results, the same evaluation criteria have been used. The parameters of the framework have been tuned using configuration 4 in Table 2, with a slight change in the way of sampling. Actually, to keep the same number of points for each example, parts are sampled with a fix number of points (1 million points). Viewpoints are also fixed for HPR by freezing one viewpoint on every face of the CAD part bounding box (front, back, top, bottom, right and left). Performing HPR from these six viewpoints ensures maximum coverage of point cloud with less viewpoints. Level of noise and outliers are taken same for these new examples to see the resulting behavior. Due to the different sizes of the parts and assemblies, the level of noise and outlier may affect the results differently. This can be observed in the Fig 10 where more noise can be seen on the clutch plate as compared to the connecting rod which has larger dimensions. Actually, in real scanning, smaller parts are difficult to handle as compared to bigger parts due to level of accuracy of laser scanners.

Table 5 represents the result of 9 examples shown in Fig 10. It can be seen that number of points have been reduced from 1 million, as there are many empty spaces in the resulting cloud due to HPR. It is not necessary that these six viewpoints will always cover the whole cloud, for full coverage we

may need many viewpoints as per complexity of CAD parts and assemblies. The values of coverage ratio also vary from product to product due to topologic configurations.

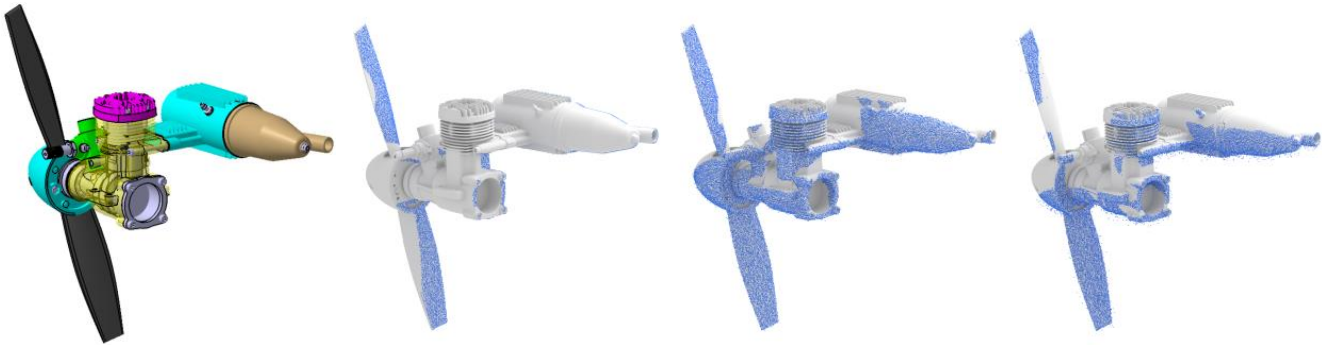


Figure 9: CAD model of a RC engine reversed engineered with configurations 1 to 3.



Figure 10: Results of the virtual Reverse Engineering process applied on CAD models following configuration 4 of Table 2, from left to right and top to bottom: clutch plate, connecting rod, clutch plate assembly, radial engine rod, hand wheel, wheel assembly, vice, machine element and stop valve assembly.

As already explained coverage ratio corresponds to the ratio between the area of the final triangle mesh generated from the point cloud and the overall area of the initial CAD model, so it can never exceed 100 percent value. Due to same configuration of parts deviations are very close to each other. Deviations are calculated as mean distance of points in resulted point cloud to the mesh part.

Models	Parts (#)	Con-fig.	Points (#)	Deviation (mm)	Coverage wrt wrapping (%)	Coverage wrt assembly (%)
Clutch plate	1	4	525,822	0.0347 (1.98)	73.5	73.5
Hand wheel	22	4	945,190	0.0185 (3.11)	90.6	42.9
Machine element	14	4	873,182	0.0195 (3.91)	82.6	47.7
Radial engine rod	1	4	791,716	0.0196 (3.58)	79.1	79.1
Clutch plate assembly	36	4	939,037	0.0190 (4.75)	89.7	23.9
Stop valve assembly	35	4	880,576	0.0192 (2.77)	87.2	48.5
Connecting rod	6	4	703,292	0.0196 (5.56)	68.3	59.4
Wheel assembly	6	4	748,718	0.0190 (2.74)	71.2	66.8
Vice	8	4	895,662	0.0182 (2.76)	90.3	62.9

Table 5: Results of the virtual Reverse Engineering process applied on CAD models according to the fourth scanning configuration of Table 2.

4.3 Comparison with real-scanned point clouds

After performing tests on different examples, a comparison has been also carried out to analyze the deviations between real-scanned and as-scanned point clouds. In this way, the parameters of the proposed approach have been tuned so that the deviations match. ROMER Absolute Arm 7520 SI (7 axis and 2m volume, absolute encoders, RS1 laser sensor 30000pts / s, volumetric accuracy of 61 μ m) was used for data acquisition of the sonotrode (Fig 1). The point cloud obtained from the real scanning of the sonotrode has been compared with the point cloud generated with our method. Recorded deviation values were used further to tune the parameters of proposed approach (e.g. noise, sampling density and level of outliers). After series of tests and adjustments in different parameters, one set of parameters resulted with a minimized deviation. The values of those tuned parameters are summarized in Tab. 6. These parameters correspond to the defaults values that can be directly specified when the user chooses to simulate such a ROMER Absolute Arm 7520 SI. Of course, for a different device, the values could be different.

Steps	Parameters	ROMER Absolute arm 7525 SI
Wrapping	Grain	0.1mm
	wrapCoverage	0
Sampling	densitySamp	10%

HPR	nbVWPts	6
	Merging	1
Misalign- ment	maxAngle	0°
	Center	Default
Noise	dirNoise	Normal
	levelNoise	0.006
Outliers	densityOut	0.1%
	levelOut	0.005

Table 6: Default values of the framework's parameters so that the generated as-scanned point clouds fit the ones obtained by a ROMER Absolute Arm 7520 SI.


5 CONCLUSION

This paper has introduced a new virtual Reverse Engineering technique able to generate as-scanned point clouds from CAD models. The method is very fast when compared to the traditional Reverse Engineering process. It does not require any tedious and time-consuming post-processing steps. It is controlled by several parameters which values can be used to insert artifacts commonly encountered when dealing with real acquisition devices. This technique has been tested on several examples from single parts to assemblies. Different configurations are used to represent the variety in data acquisition as if the point clouds are obtained from different scanners. Parameters used for generating these results can be fine-tuned with the parameters of scanners for sensitivity and accuracy. The CAD assemblies considered here are perfect (perfectly fitted interfaces, sharp edges etc) which of course is not the same for product that has been manufactured. Effectively this is a limitation that the digital mockups of CAD assemblies used for this framework do not contain defects caused by manufacturing processes. Similarly, type of material is also not directly considered in this framework but rather indirectly through other parameters introducing artifacts which can be ascribed to surface properties. The proposed technique can be used to develop database of point clouds for educational and research purposes. This is a modular based approach and new capabilities can also be added to enrich the data. The next steps concern the definition of pre-defined configurations of the parameters so as to help the user instantiating them, the labeling of the point clouds while developing mechanisms able to capture and propagate the information from the CAD models, the generation of huge databases of as-scanned point clouds to be used for Artificial Intelligence applications.

Jérémy Montlahuc, <http://orcid.org/0000-0003-2441-6812>
 Ghazanfar Ali Shah, <http://orcid.org/0000-0003-3740-9945>
 Arnaud Polette, <http://orcid.org/0000-0002-8572-6454>
 Jean-Philippe Pernot, <http://orcid.org/0000-0002-9061-2937>

REFERENCES

- [1] Bagci, E: Reverse engineering applications for recovery of broken or worn parts and re-manufacturing, Three case studies, *Advances in Engineering Software*, 40(6), 2009, 407-418 <https://doi.org/10.1016/j.advengsoft.2008.07.003>
- [2] Berger, M.; Tagliasacchi, A.; Seversky, L.; Alliez, P.; Guennebaud, G.; Levine, J. A.; Sharf, A.; Silva, C. T.: A Survey of Surface Reconstruction from Point Clouds, *Computer Graphics Forum*, 36(1) 2017, 301-329. <https://doi.org/10.1111/cgf.12802>
- [3] Cloudcompare. <http://www.cloudcompare.org>
- [4] Dassault Systèmes, <https://www.3ds.com>
- [5] Huang, J.; You, S.: Point cloud labeling using 3D Convolutional Neural Network, *International Conf. on Pattern Recognition*, 2016, 2670-2675. <https://doi.org/10.1109/ICPR.2016.7900038>
- [6] Katz, S.; Tal, A.; Basri, R.: Direct Visibility of Point Sets, *Association for computing machinery Transactions on Graphics*, 26(3), 2007, 24. <https://doi.org/10.1145/1276377.1276407>

- [7] Khalfaoui, S.; Aigueperse, A.; Seulin, R.; Fougerolle, Y.; Fofi, D.: Fully automatic 3D digitization of unknown objects using progressive data bounding box, *Three-Dimensional Image Processing and Applications II*, 2012, 829011. <https://doi.org/10.1117/12.909164>
- [8] Kumar, A.; Jain, P. K.; Pathak, P. M: Reverse Engineering in  *Product Manufacturing*, 2013, 665-678. <https://doi.org/10.2507/daaam.scibook.2013.39>
- [9] Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *IEEE Conf. on Computer Vision and Pattern Recognition*, 1(2), 2017, 77-85.
- [10] Sun, X.; Rosin, P. L.; Martin, R. R.; Langbein, F. C.: Noise in 3D Laser Range Scanner Data, *Shape Modeling and Applications*, 2008, 37-45.

1.2 Conclusion

This chapter introduced a new as-scanned point cloud generation technique to virtually generate a large dataset of point clouds to validate and improve the new reverse engineering technique in the following chapters. The method does not require any post-processing (e.g. cleaning, filtering, registration, simplification, meshing) of point clouds and it can also handle the generation of several point clouds in a single batch. Moreover, this technique is independent of the complex phenomena generated by, and/or between, the acquisition device (e.g. type, control parameters), the digitalized object (e.g. material, color, shape, size), the operator (e.g. acquisition strategy, experience), and the environment (e.g. light, temperature, vibration). The tool has been tested on several single and assembly parts for the generation of point clouds that are further tested by the new reverse engineering technique. Even if this method has been originally designed in the reverse engineering context, new applications are now explored and notably the ones related to machine learning that requires a large database on which to learn. For example, segmentation and classification of point clouds based on artificial intelligence require access to large datasets which may also have to be labeled when considering supervised learning techniques. For a given acquisition device, the parameters of our proposed method can be tuned to define “default parameters” (e.g. level of noise, tessellation accuracy, level of outliers etc.) so that the as-scanned strategy generates point clouds as similar as possible to the ones obtained with the acquisition devices. One of the advantages of this tool is that the results can easily be manipulated by varying the input parameters. For instance, the variation of the noise level allows a better understanding of fitting behaviors for a different level of noise.

1.2. CONCLUSION

CHAPTER



Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies

Contents

2.1	Introduction	30
2.2	Conclusion	52

2.1 Introduction

In connection with the overall objective of the thesis illustrated in Figure 1, this chapter introduces the technique of Simulated Annealing-based fitting of CAD models to their respective point clouds. The point cloud used for the fitting in this chapter come either from real scanners or from CAD models through the as-scanned point cloud generation strategy (Chapter 1). The fitting technique relies on the idea of using parametric CAD models whose parameters are to be optimized using Simulated Annealing algorithm to minimize the deviations between the CAD models to be fitted and the point cloud.

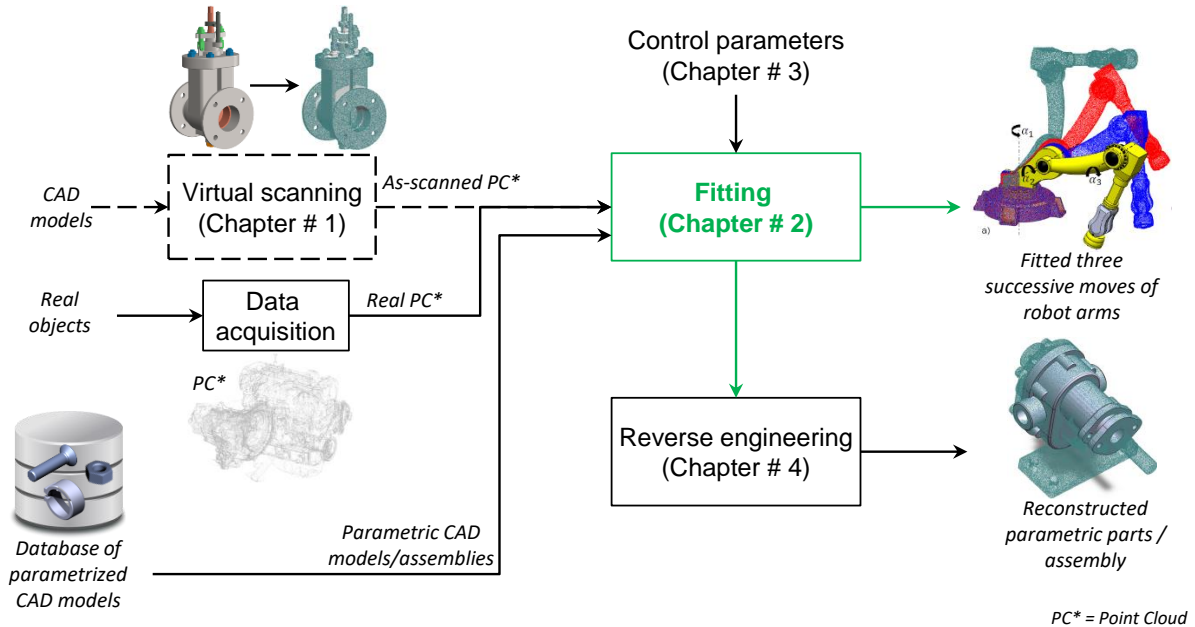


Figure 1 – Fitting technique supporting the new reverse engineering to reconstruct mechanical parts/assembly models.

The proposed approach is modular, and different modules perform different tasks (e.g. distance computation, update of CAD geometry, constraints satisfaction, registration, etc.). For instance, the parameters of the CAD model to be optimized can also be related to the assembly constraints e.g. distance between the two parts. The proposed method is supported by two nested loops: the first for the segmentation of point cloud for assembly fitting; the second for the grouping of parameters to be optimized by SA. Several examples related to the single and assembly CAD models are validated on both local and global fitting cases. To strengthen the fitting accuracy, the approach is supported by the Iterative Closest Points (ICP) algorithm that works throughout the optimization process

2.1. INTRODUCTION

for the orientation and positioning of the CAD models. Multiple parts are fitted individually and simultaneously in the point clouds using the proposed method. This chapter is a prerequisite for the development of the new reverse engineering technique in Chapter 4.

2.1. INTRODUCTION



Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies

Ghazanfar Ali Shah^{1,2,3} · Arnaud Polette¹ · Jean-Philippe Pernot¹ · Franca Giannini² · Marina Monti²

Received: 10 July 2019 / Accepted: 29 January 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

This paper introduces a new fitting approach to allow an efficient part-by-part reconstruction or update of editable CAD models fitting the point cloud of a digitized mechanical parts' assembly. The idea is to make use of parameterized CAD models whose dimensional parameters are to be optimized to match the acquired point cloud. Parameters may also be related to assembly constraints, e.g. the distance between two parts. The optimization kernel relies on a simulated annealing algorithm to find out the best values of the parameters so as to minimize the deviations between the point cloud and the CAD models to be fitted. Both global and local fitting are possible. During the optimization process, the orientation and positioning of the CAD parts are driven by an ICP algorithm. The modifications are ensured by the batch calls to a CAD modeler which updates the models as the fitting process goes on. The modeler also handles the assembly constraints. Both single and multiple parts can be fitted, either sequentially or simultaneously. The evaluation of the proposed approach is performed using both real scanned point clouds and as-scanned virtually generated point clouds which incorporate several artifacts that could appear with a real scanner. Results cover several Industry 4.0 related application scenarios, ranging from the global fitting of a single part to the update of a complete Digital Mock-Up embedding assembly constraints. The proposed approach demonstrates good capacities to help maintaining the coherence between a product/system and its digital twin.

Keywords CAD assembly models · Digital twin · Constrained fitting · Registration · Simulated annealing · ICP · As-scanned point clouds

1 Introduction

Today, the needs to reconstruct or update 3D information related to real-world objects, products, systems, buildings, environments, terrestrial surfaces and even human beings have become mainstream. This topic turns out to be of primary interest in the scope of the fourth industrial

revolution, commonly known as Industry 4.0, and for which the demand for access to and use of up-to-date digitized information related to the digital twin of a system has raised [1]. The increased interest in 3D digitization, combined to the emergence of low-cost devices have certainly speeded up the development and spreading of new techniques in many different application domains, ranging from facial reconstruction and comparison on smartphones to complex reconstructions for mechanical engineering or architectural applications. Depending on the application domain, a wide variety of scenarios can be foreseen, and the needs range from the acquisition and treatment of incomplete point clouds, to the full reconstruction or update of CAD assembly models potentially composed of several components and parts. In product development, the objective is to reconstruct or to update CAD parts, CAD assembly models and even full Digital Mock-Up (DMU) which can then be exploited at different stages of the Product Development Process (PDP). For instance, the reconstructed or updated 3D models can then be exploited to simulate products' and systems' behavior, so

✉ Ghazanfar Ali Shah
ghazanfar_ali.shah@ensam.eu
Jean-Philippe Pernot
jean-philippe.pernot@ensam.eu

¹ Arts et Métiers Institute of Technology, LISPEN, HESAM Université, F-13617 Aix-en-Provence, France

² Istituto di Matematica Applicata e Tecnologie Informatiche "Enrico Magenes", CNR Via De Marini 6, 16149 Genoa, Italy

³ DIME-Dipartimento di Ingegneria meccanica, energetica, gestionale e dei trasporti, Università degli Studi di Genova, Genoa, Italy

as to understand the origin of some failures, to define mitigation plans and come up with enhanced performances. It can also be used to check assembly specifications of products and systems which would not have been disassembled before scanning, as well as many more scenarios. Actually, such an approach can greatly enhance the capacity of companies to develop more competitive products, while reducing the development times and costs [2]. In the scope of the Industry 4.0, the idea is to be able to update an as-is DMU so as to maintain the coherence between a physical system and its digital twin. This could be very helpful to get access to real-time information regarding for instance the state of a machine or of a production line. At the end, this would allow more accurate and faster simulations and analyses as well as more proactiveness and agility for decision-making.

Unfortunately, current approaches do not fully meet the above-mentioned requirements for high-level reconstruction and update tools able to consider the information at the level of the parts, assemblies and underlying semantics. Today, the reconstruction of CAD models from point clouds follows a more or less manual, cumbersome, and time-consuming patch-by-patch fitting strategy during which engineers have to face many issues: pre-processing of the data, segmentation of point clouds, decomposition in patches, fitting of primitives, trimming and stitching of the resulting surfaces. At the end, manifold B-Rep models are obtained but cannot be easily modified as they are considered as dead models which do not rely on real building trees [3]. This is an important limitation of the existing techniques, which prevents efficient reconstructions or updates of consistent and editable CAD models. Actually, despite this effervescence around the development of new reconstruction algorithms, very few address the challenging problem of reconstructing or updating CAD models that could then be edited and more successfully exploited in the downstream stages of the PDP. This is true when considering single parts and it is even truer for the assemblies of several parts potentially subjected to assembly constraints. Again, being able to check as-built and as-is models and tolerances as well as to analyze accessibility issues for maintenance planning without disassembling the components can drastically speed up the reverse engineering process and thus improve the performance of the PDP. More generally, being able to keep track of the evolutions of a product/system through its digital twin is of foremost importance to support decision-making and to evolve towards more optimized and autonomous products and systems. This requires a paradigm shift.

This paper introduces a new part-by-part fitting approach to circumvent those issues and allow a more global and efficient reconstruction or update of editable CAD models fitting the point cloud of a digitized mechanical parts' assembly. Depending on the adopted scenario, assembly constraints can also be specified and maintained during the

fitting process. The idea is to directly make use of parameterized CAD models whose parameters are to be optimized. Parameters may also be related to assembly constraints, e.g. the distance between two parts. The proposed fitting approach relies on an optimization kernel which makes use of a simulated annealing (SA) algorithm to find out the best values of the parameters so as to minimize the deviations between the point cloud and the CAD models to be fitted. Both global and local fitting are possible. During the optimization process, the orientation and positioning of the CAD parts are driven by an ICP algorithm. The consistency of the CAD models is ensured by calling a modeler, which updates the CAD models as the fitting process goes on. This reduces the risk to get for instance badly fitted surfaces, gaps, overlapping surfaces, self-intersections. The modeler also handles the assembly constraints when specified between the CAD parts. Both single and multiple parts can be fitted, either sequentially or simultaneously.

The contribution is threefold: (1) a new part-by-part fitting framework allows the fitting of parameterized and editable CAD models to point clouds of digitized mechanical parts assemblies; (2) both local and global fitting can be performed thanks to a segmentation strategy controlled by a simulated annealing optimization algorithm; (3) the method can fit single parts as well as multiple parts optionally constrained together with assembly constraints and fitted simultaneously. The proposed approach demonstrates good capacities to help maintaining the coherence between a product or system and its digital twin.

The paper is organized as follows. Section 2 reviews the related works and positions the fitting technique with respect to state-of-the-art approaches. The proposed framework is then introduced in Sect. 3 and the details of the building blocks are given in Sect. 4. The approach is then tested and validated on several test cases, including both global and local fitting configurations, on single parts as well as on assemblies of parts subjected to assembly constraints (Sect. 5). Section 6 ends this paper with conclusions and perspectives.

2 Related works

The literature is plenty of more or less sophisticated reconstruction techniques from point clouds [4]. Here, a focus is put on fitting methods of interest to reconstruct or update CAD models. To classify the existing approaches, it is important to stress that efficient reconstruction algorithms use priors, i.e. assumptions made by algorithms to combat imperfections in the point cloud and to eventually focus what information about the shape is reconstructed. Without prior assumptions, the reconstruction problem is ill-posed, i.e. an infinite number of solutions can satisfy the fitting

problem. Of course, when addressing update issues, prior assumption is satisfied straightforwardly since it is assumed that the model to be updated exists. More precisely, the techniques which can help reconstructing or updating CAD models from point clouds can make use of three main priors: the geometric primitive prior, the global regularity prior and the data driven prior.

The *geometric primitive* prior is adopted in several techniques which try to fit basic geometric primitives in point clouds. Among the existing techniques, Fischler et al. have introduced the well-known random sampling consensus (RANSAC) paradigm to extract shapes by randomly identifying minimal sets from the point cloud and constructing corresponding shape primitives [5]. Schnabel et al. have proposed an automatic algorithm to detect basic shapes in unorganized point clouds [6]. Their method is based on RANSAC paradigm and performs a random sampling to detect planes, spheres, cylinders, cones and torii. For models with surfaces composed of these basic shapes only, e.g. CAD models, the resulting representation solely consists of shape proxies. This method has been extended not only to fill in the gaps between the detected primitives but also to synthesize plausible edges and corners generated from the intersections of basic primitives [7]. The method can be applied on point clouds of mechanical assemblies but it is unable to identify the parts properly saying. Bey et al. [8] have proposed a method to reconstruct CAD models from 3D point clouds, assuming that an a priori CAD model, roughly similar to the scene to reconstruct, is given. Their work is actually limited to cylindrical shapes and does not work on complete CAD models. Lari et al. have introduced an approach for the identification, parameterization, and segmentation of planar and linear/cylindrical features from laser scanning data, while considering the internal characteristics of the input point cloud, i.e. local point density variation and noise level in the dataset [9]. Their method is limited to very simple primitives and the resulting model is not watertight. Clustering approaches can also be used to detect geometric primitives in point clouds. To this aim, Attene et al. have proposed a hierarchical face clustering algorithm for triangle meshes based on fitting primitives belonging to an arbitrary set [10]. Their method is particularly efficient and is completely automatic which can be interesting in the reverse engineering context. It generates a binary tree of clusters, each of which fitted by one of the primitives employed. However, their method does not perform well on incomplete scanned data and there is no CAD model reconstruction.

The *global regularity* prior deals with high-level properties such as symmetries, structural repetitions, and canonical relationship [11]. 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. [12] 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 [6], 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 correspond to a man-made engineering object consisting of basic primitives, possibly repeated and globally aligned under common relations. 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 [13]. 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. Finally, 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 DMU. Such an ability would, however, be of great interest to update digital twins, and thus, to answer some Industry 4.0-related needs.

The *data driven* prior makes use of existing objects or object parts to be fitted to the point clouds in a rigid or non-rigid manner. This is much more studied in the context of scene understanding than for the reconstruction or update of mechanical parts assemblies. For instance, Nan et al. [14] have proposed an interesting approach for indoor scene understanding. The method is based on initial random selection and iterative region growing with increasing classification likelihood. The scene is then reconstructed by deforming template models to fit the classified points. However, this technique allows for global fitting only and the adopted templates are far more simple than real industrial CAD models which integrate much more features. Assembly constraints are also not considered. Anyway, in the CAD domain, Liu Ip et al. have proposed a new approach to retrieve a CAD model in a point cloud and to align it using a transformation based on principal components analysis [15]. Here, the parameters of the CAD model are not modified so as to best fit the point cloud. The rigid transformation could also be obtained using an efficient variant of the ICP (Iterative Closest Point) algorithm [16]. Rabbani et al. have compared several techniques to directly fit CSG objects to point clouds [17]. Their ICT (Iterative Closest Triangle-point) algorithm is able to identify the values of some free parameters as well as to maintain the relationships between some other parameters. The algorithm can perform global fitting which limits its use when considering mechanical parts' assemblies. Buonamici et al. [18] have introduced a template-based technique for the reverse engineering of mechanical parts. Here, a CAD template is fitted upon the mesh generated from the point cloud, optimizing its dimensional parameters and position/

orientation by means of particle swarm optimization algorithm. In their approach, the reference mesh is subdivided into N meshes (i.e. N single features) through a segmentation process with the help of a commercial RE software, and afterwards, each segmented region is then registered to the corresponding surface of the CAD model. This method allows for a global fitting of the CAD template to the reference mesh. Unfortunately, the method focuses on global fitting of parts, and neither the reconstruction nor update of CAD assembly models is considered. Wang et al. [19] have designed a framework to create 3D models from the boundary surface meshes of industrial parts. It exploits a divide-and-conquer strategy to construct all primitive features of parts. According to the geometric and topological relationships among features, some modeling operations are performed to obtain the final model. Here again, the result is a dead B-Rep model which can be hardly modified in the downstream stages of the PDP. Another semi-automated approach is introduced by Stark et al. [20] for the reconstruction of 3D assembly models through the process of segmentation, part identification and structure identification. Point cloud data obtained from scanning are used for finding parts from database and retrieved parts are used for reconstruction of assembly model. Bénire et al. [21] have introduced an automatic process for the reverse engineering of models from 3D meshes. However, the method works on meshes obtained by mechanical object discretization. Finally, Xu et al. [22] have developed a modeling strategy to reconstruct a mechanism from multi-view images. It uses an interactive part modeling step to draft the parts which are then fitted to the mechanism point cloud. Parts are then aligned using an optimization step and the motion parameters can be estimated using a pre-recorded video clip of the mechanism motion. This method is interesting as it does not need to use a parameterized model as input; however, the segmentation properly speaking is left to the user who specifies the correspondences between the parts and the point cloud.

As conclusion, even if some of the previously discussed methods can be of interest to support the fitting process of mechanical parts, they still suffer from several limitations. First, mostly single CAD parts are reconstructed,

and very few attention is paid to the direct reconstruction or update of CAD assembly models from digitized part assemblies. Indeed, existing methods mostly work at the level of the parts, and they require the assembly to be disassembled. This is incompatible when considering the needs for checking and processing as-is DMUs, and especially in the context of the Industry 4.0 where the digital twins should follow as much as possible the evolution of the physical products/systems, without disassembling them. Second, the reconstructed CAD models cannot be easily modified in the downstream stages of the PDP as they often correspond to dead B-Rep models. These are the limitations to overcome with the proposed fitting framework. In line with the Industry 4.0-related scenarios identified and discussed in the introduction, the proposed method exploits the assumption that parameterized CAD models to be fitted are a priori identified.

3 Overall framework

This section introduces the overall framework of the simulated annealing-based fitting technique (Fig. 1). The details of the different modules are provided in Sect. 4. Starting from a real-life mechanical parts' assembly, a point cloud PC_0 is first acquired using for instance a laser scanner. One or several CAD models to be fitted in the point cloud are then selected, being \mathcal{M}^κ the κ -th one (with $\kappa \in \{1, \dots, N_m\}$, and N_m the number of models to be fitted). CAD models can optionally be part of an assembly structure \mathcal{A} , potentially composed of subassemblies, and their fitting can be performed either sequentially or simultaneously. Here, it is assumed that the CAD models are parameterized at the level of the parts (e.g. lengths, diameters and angles parameterizing the features) and/or at the level of the assemblies (e.g. lengths and angles parameterizing the relative positions and orientation of the parts the ones with the others). Both types of parameters can be considered as numerical variables x_k of the fitting process. Each CAD model may also contain additional internal constraints (e.g. equations linking parameters of the part, constraints between geometric

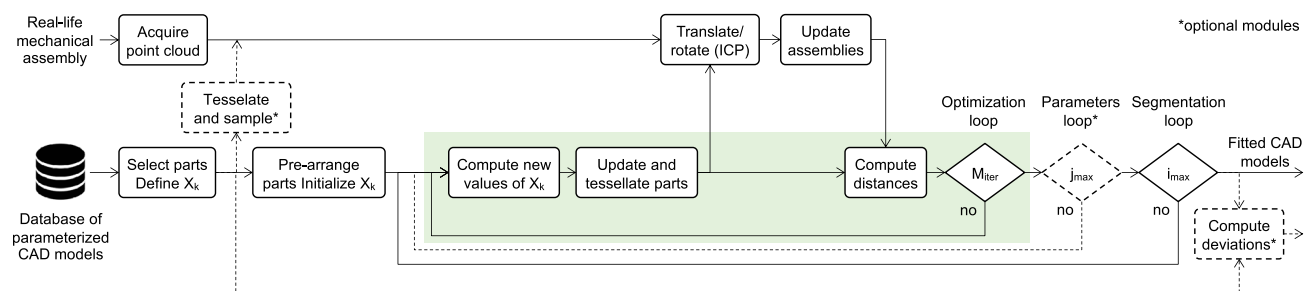


Fig. 1 Overall framework to fit parameterized CAD assembly models to point clouds of digitized mechanical part assemblies

entities as for instance perpendicularity and parallelism), and so does each assembly (e.g. coaxiality of two axes, contact between two parts, equations between the parts' parameters). These internal constraints do not take part to the optimization process, and are directly handled by the CAD modeler in charge of updating the assemblies and CAD models each time their numerical parameters change. The numerical parameters can optionally be gathered in several groups \mathcal{G}_j , with $j \in \{1, \dots, j_{\max}\}$, corresponding to the level of details they are related to. The parts are pre-arranged within the point cloud; this initializes the values x_k^0 of all the numerical variables. The iterative fitting process then starts following three nested loops (Fig. 1):

- the *segmentation loop* aims at segmenting the initial point cloud PC_0 so as to consider only the points in the surrounding of the N_m models to be fitted, and thus to allow for local fitting. This iterative process is performed i_{\max} times. During the i -th loop, with $i \in \{1, \dots, i_{\max}\}$, a threshold \mathcal{E}_i^x is computed for each CAD model ($\kappa \in \{1, \dots, N_m\}$). It is used to get the surrounding point cloud PC_i^κ , while cropping PC_{i-1}^κ obtained during the previous segmentation loop. Thus, the number of points to be considered in the optimization loop progressively reduces step after step. For the first loop ($i = 1$), each \mathcal{E}_1^x is initialized from the oriented bounding box of the κ -th pre-arranged CAD model. These thresholds are used to initialize the segmentation and to generate the various PC_1^κ from the initial point cloud PC_0 and from the position of the pre-arranged CAD models. In case of global fitting, the segmentation loop is not used and the point cloud PC_0 is considered as a whole all along the fitting process;
- the *parameters loop* is optional and it is executed only if more than one group of parameters ($j_{\max} > 1$) is defined. For each step of the segmentation loop, the parameter loop treats the j_{\max} groups one after the other. Each group \mathcal{G}_j contains one or more parameters $x_{j,k}$ to be optimized in the optimization loop;
- the *optimization loop* is the core of the proposed approach. It is based on a simulated annealing algorithm which iteratively modifies the numerical parameters of the CAD models until they perfectly fit their respective cropped point clouds PC_i^κ according to a stop criterion. Here, the algorithm works on a given group \mathcal{G}_j containing a restricted set of parameters $x_{j,k}$ to be optimized. During this loop, several steps are to be performed. First, the SA algorithm identifies new parameters' values $x_{j,k}(t)$, and the CAD models are then updated accordingly. The update is left to the CAD modeler which ensures the consistency of the updated B-Rep models and manages the internal constraints defined in the building trees of the parts. The updated CAD models

are then tessellated on one hand, and used as input of an ICP algorithm on the other hand. The ICP algorithm is used to compute a homogeneous transformation matrix so that the updated CAD models align with their respective PC_i^κ . The CAD modeler then updates the assemblies so as to satisfy the possibly defined assembly constraints between the parts. Thus, the position and orientation of the parts are not considered directly as variables of the optimization process. The distances between the updated CAD models and their correspondent cropped point clouds PC_i^κ can now be computed and the result serves as the objective function to be minimized during the optimization. The optimization loop stops when a max number of iteration M_{iter} without any change (up to a given accuracy ϵ_{iter}) is reached, otherwise the SA algorithm goes on with the computation of new values $x_{j,k}(t+1)$. Once the optimal values of the parameters have been found, the algorithm goes back to the parameters loop if $j < j_{\max}$, otherwise it goes back to the segmentation loop so as to start a new segmentation if $i < i_{\max}$.

At the end of the loops, one or several fitted CAD models are obtained. Depending on the adopted scenario and needs, this fitting process can be repeated for other parts of the digitized assembly. Finally, since the fitted CAD models are editable, the user can still modify them while rounding for instance the values of the optimized parameters. Moreover, the final cropped point clouds $PC_{i_{\max}}^\kappa$ correspond to partial segmentations of PC_0 reflecting the fitted parts.

For experimentation and validation purposes, the scanning of a real-life mechanical assembly can be optionally bypassed (dashed lines of Fig. 1) by the automatic generation of an as-scanned point cloud using the virtual acquisition technique of Montlahuc et al. [23]. In this case, the initial CAD models to be fitted are known, and the values of their parameters can be compared with the ones resulting from the proposed fitting strategy. The virtual scanning technique is briefly introduced in Sect. 5 as it is used to benchmark and validate the fitting algorithm.

4 Fitting framework modules

This section introduces the technical aspects underlying the fitting of one or several CAD models \mathcal{M}_m in the point cloud PC_0 of a digitized parts' assembly. CAD models can optionally be part of an assembly structure \mathcal{A} , potentially composed of subassemblies, and constrained with assembly constraints. For sake of clarity, the successive steps are illustrated on the fitting of a single flange in the point cloud of a digitized valve (Fig. 2). The point cloud is composed of 1204k points. Section 5 presents and discusses more results

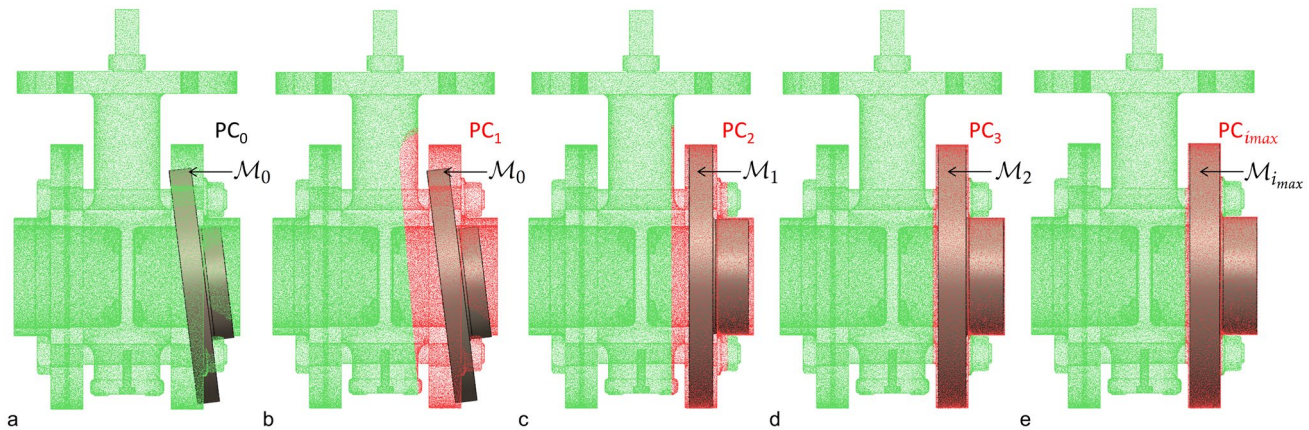


Fig. 2 Fitting the parameterized CAD model of a flange in the point cloud of a digitized valve assembly ($N_m = 1$): **a** coarse pre-arrangement in the initial point cloud PC_0 ; **b–d** evolution of the cropped

point clouds PC_i during the segmentation loops; **e** final fitted flange $M_{i_{max}}$ and associated segmentation $PC_{i_{max}}$

to cover various fitting scenarios involving several parts and assembly structures.

4.1 CAD models' selection and pre-arrangement

Once the parts' assembly has been digitized, a point cloud PC_0 is available; the first step is to select the CAD models to be fitted. The way this is performed depends on the adopted fitting scenario. The models can come from an existing database available in the company, when considering for instance the need to either update the DMU of a system/product or control its assembly. They can also be roughly and rapidly sketched and parameterized starting from scratch using the CAD modeler. Designers can also make use of other existing databases which could be browsed using ad hoc assembly retrieval approaches [24].

Each of the N_m selected CAD model, in the following indicated as M^κ , is defined by a set of control parameters $p_{k'}^\kappa$, $k' \in \{1, \dots, N_p^\kappa\}$, which correspond to either dimensions or angles. Additional numerical parameters $a_{k'}$, $k' \in \{1, \dots, N_a\}$ may also be considered if the CAD models are part of an assembly structure \mathcal{A} involving assembly constraints also parameterized by dimensions or angles. Both types of parameters are considered as numerical variables x_k to be optimized during the optimization loop. The assembly structure and the CAD models may also include built-in constraints (e.g. contact between parts, coaxiality, symmetry, parallelism, relationships between parameters) to be maintained during the modifications. The updates of the CAD models and assembly structure are left to a CAD modeler which ensures the consistency of the B-Rep model all along the optimization process. Thus, both the assembly structure \mathcal{A} and its

associated CAD models M^κ can be seen as the result of several generation functions g^* described in the assembly and building trees and so that:

$$\begin{cases} \mathcal{A} = g^a(a_1, \dots, a_{N_a}) \\ M^\kappa = g^\kappa(p_1^\kappa, \dots, p_{N_p^\kappa}^\kappa), \quad \forall \kappa \in \{1, \dots, N_m\} \end{cases} \quad (1)$$

Optionally, the control parameters can be split in several groups \mathcal{G}_j , $j \in \{1, \dots, j_{max}\}$, according to the level of details to which they correspond. The parameters of the group \mathcal{G}_j are denoted $p_{j,k}$, $k \in \{1, \dots, N_{pj}\}$, and the numerical variables for the optimization loops are the $x_{j,k} = p_{j,k}$. Actually, using a three-level decomposition turns out to be a good trade-off considering commonly adopted CAD modeling strategies ($j_{max} = 3$): \mathcal{G}_1 groups the parameters of the assembly structure as well as the parameters driving the structural features (e.g. pads, pockets, revolutions), \mathcal{G}_2 gathers together the parameters of the detail features (e.g. holes, ribs), and \mathcal{G}_3 is concerned by the parameters used to finalize the CAD model (e.g. fillets, chamfers). When considering the fitting of several CAD models, all their parameters are mixed up within those three groups. Thus, when referring to those groups, the upper indices κ is no more used. Moreover, to simplify the writing, this upper indices will also not be used in case a single part is to be fitted ($N_m = 1$).

For example, Fig. 3 shows the parameters of the single flange ($N_m = 1$) to be locally fitted in PC_0 as shown on Fig. 2. It is defined by eight control parameters ($N_p = 8$): radii r_i with $i \in \{1, \dots, 6\}$, heights h_1 and h_2 . It also contains symmetry constraints to be maintained by the CAD modeler. As only one single flange is to be fitted, there are no assembly constraints ($N_a = 0$). The three groups can then be defined as follows:

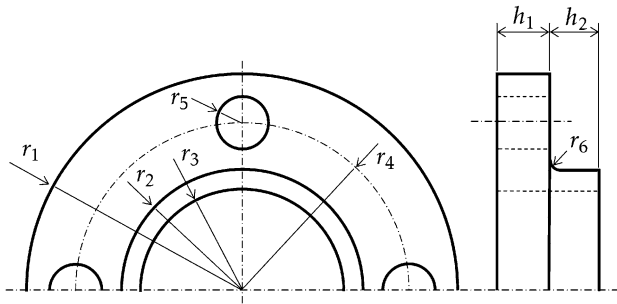


Fig. 3 Definition of the eight parameters controlling the shape of the half-flange that is fitted in the point cloud of a digitized assembly as depicted on Fig. 2

$$\begin{aligned}
 \mathcal{G}_1 &= \{r_1, r_2, r_3, h_1, h_2\} : p_{1,1} = r_1, p_{1,2} = r_2, \\
 p_{1,3} &= r_3, p_{1,4} = h_1 \text{ and } p_{1,5} = h_2, \\
 \mathcal{G}_2 &= \{r_4, r_5\} : p_{2,1} = r_4 \text{ and } p_{2,2} = r_5, \\
 \mathcal{G}_3 &= \{r_6\} : p_{3,1} = r_6.
 \end{aligned} \tag{2}$$

The CAD models and the assembly structure are then pre-arranged in the point cloud PC_0 ; this initializes the values x_k^0 of all the parameters, and thus it initializes the pre-arranged CAD models \mathcal{M}_0^k and assembly structure \mathcal{A}_0 . This step depends on the adopted fitting scenario. For instance, it can be performed using pre-identified primitives to align certain axes of the CAD models with axes obtained using for instance RANSAC paradigm [6]. The initial values of the control parameters x_k^0 can also be tuned in a coarse manner or using global scaling factors. Of course, as for any numerical method, the idea is to start as close as possible to the final solution to avoid local minimums. This step is illustrated on Fig. 2a where a coarse pre-arrangement of the flange has been performed.

4.2 Segmentation loop

From this pre-arrangement, if a local fitting is to be considered, the algorithm crops the initial point cloud PC_0 to get the subsets PC_1^k associated to the N_m models to be fitted. For the κ -th model, the cropping is driven by an initial threshold \mathcal{E}_1^k . All the points of PC_0 which have a distance to \mathcal{M}_0^k greater than \mathcal{E}_1^k are cropped (Fig. 2b). By default, \mathcal{E}_1^k is set up to 10% of the diagonal of the pre-arranged part oriented bounding box. This percentage corresponds somehow to the level of confidence in the pre-arrangement, thus the default value can be adjusted. The cropping is done in the same way for each model \mathcal{M}^k , but with different \mathcal{E}_1^k values. The cropped PC_1^k are then used to optimize the parameters of all the groups \mathcal{G}_j in the parameters and optimization loops. Those two loops end up with the definition of the \mathcal{M}_1^k fitting the PC_1^k . The process goes on this way with

successive values $\mathcal{E}_i^k, i \in \{2, \dots, i_{\max}\}$ used to crop all the points of the PC_{i-1}^k which have a distance to the \mathcal{M}_{i-1}^k greater than \mathcal{E}_i^k , and thus define the PC_i^k used to generate the \mathcal{M}_i^k . Here, the \mathcal{E}_i^k are defined according to the max distance between the PC_{i-1}^k and the \mathcal{M}_{i-1}^k such that

$$\mathcal{E}_i^k = \omega_s^k \times \max \{ d(\mathcal{M}_{i-1}^k, PC_{i-1}^k[\tau]) : \tau \in [1..n_{i-1}^k] \}, \tag{3}$$

wherein n_{i-1}^k is the number of points in PC_{i-1}^k and ω_s^k is a segmentation weight empirically set up to 10%.

The segmentation loops are illustrated on the example of Fig. 2. Figure 2c shows \mathcal{M}_1 fitting PC_1 and used to compute \mathcal{E}_2 and PC_2 . Similarly, Fig. 2d depicts \mathcal{M}_2 fitting PC_2 and used to get \mathcal{E}_3 and PC_3 . Figure 2e shows the final fitted CAD model $\mathcal{M}_{i_{\max}}$ fitting $PC_{i_{\max}}$, with $i_{\max} = 3$.

4.3 Parameters loop

As introduced in Sect. 4.1, the parameters $x_{j,k}$ of the j -th group can refer to either parameters of the assembly structure or parameters of one or several CAD models. The parameters grouping is not performed according to the models but considering the level of details to which they correspond. Thus, the number of groups j_{\max} is limited and is not affected by the number N_m of models to be fitted.

If groups of parameters are considered ($j_{\max} > 1$), then this nested loop treats each group one by one, otherwise the parameters are treated all together in a single step. The treatment makes use of the cropped point clouds PC_i^k generated in the i -th segmentation loop. For each group $\mathcal{G}_j, j \in \{1, \dots, j_{\max}\}$, it consists in running the optimization loop to get the optimized parameters $x_{j,k}$, with $k \in \{1, \dots, N_{pj}\}$.

4.4 Optimization loop

The objective of this loop is to find out the optimal values of the parameters $x_{j,k}$ of a parameters group \mathcal{G}_j (parameters loop) so that the CAD models \mathcal{M}_i^k best fit their respective cropped point clouds PC_i^k (segmentation loop). Actually, this can be formulated as a minimization problem:

$$\min_{\substack{x_{j,k} \in \mathcal{D}_{j,k} \\ k \in [1..N_{pj}]} } E_{i,j}(x_{j,1}, \dots, x_{j,N_{pj}}) = \sum_{\kappa=1}^{N_m} d(PC_i^{\kappa}, \mathcal{M}_i^{\kappa}), \tag{4}$$

where $x_{j,k}$ are the variables, $\mathcal{D}_{j,k}$ their definition domains, \mathcal{M}_i^k the CAD models whose deviations to the cropped point clouds PC_i^k are to be minimized, and $E_{i,j}$ the energy function characterizing this overall deviation. During the successive optimization loops, the \mathcal{M}_i^k are updated by the CAD modeler using the generation functions g^* of Eq. (1) where only the subset of parameters $x_{j,k}$ is modified. As previously said,

the functions g^* incorporate built-in constraints (e.g. symmetries, relationships) which are not directly accessible in the optimization loop but can be satisfied from the building and assembly trees. The way the distances between the points of the PC_i^κ and the CAD models \mathcal{M}_i^κ are computed is explained in Sect. 4.4.4.

4.4.1 Resolution using simulated annealing algorithm

The minimization of the energy function $E_{i,j}$ is obtained using a metaheuristic algorithm able to solve the optimization problem in a large solution space. Simulated annealing (SA) algorithm has demonstrated good proficiency to find efficiently the solution from a pre-arranged configuration. It performs a global stochastic search that evolves towards local searches as the time goes on. Thus, it is particularly interesting in the present case for which the initial positions have been pre-arranged in the close vicinity of the final fitted configurations. The algorithm handles a very limited set of constraints, mainly the lower and upper bounds of the parameters values, and both the internal and assembly constraints (e.g. coincident, parallelism, coaxiality, contact) are left to the CAD modeler in charge of the updates. The use of such an algorithm is particularly interesting in the present case as the energy function to be minimized, and the geometric constraints to be satisfied, are not defined by equations but by means of black boxes combining calls to several procedures of the CAD modeler [25, 26]. Other metaheuristics have been tested but have demonstrated a lower efficiency than SA. For instance, particle swarm optimization (PSO) generates candidate solutions which can be significantly different from the initial position. This optimization strategy may result in configurations that are hard for the CAD modeler to update, and may even cause the software to crash.

Starting from an initial configuration, potentially composed of an assembly structure $\mathcal{A}(0)$ and one or several CAD models $\mathcal{M}_i^\kappa(0)$ defined by the variables $x_{j,k}(0)$ of the j -th group, the algorithm iterates on the parameters $x_{j,k}(t)$ thus creating an evolution of the current assembly structure $\mathcal{A}(t)$ and models $\mathcal{M}_i^\kappa(t)$. The optimization loop stops when a max number of iteration M_{iter} without any change of $E_{i,j}$ (up to a given accuracy ϵ_{iter}) is reached, otherwise the SA algorithm goes on with the computation of new values $x_{j,k}(t+1)$ defining an updated assembly structure $\mathcal{A}(t+1)$ and updated models $\mathcal{M}_i^\kappa(t+1)$. In this nested loop, the updates are performed while only considering the parameters of the j -th group. The way the SA control parameters are tuned is discussed in Sect. 5.2.

4.4.2 CAD models update and tessellation

The SA algorithm evolves thanks to the variations of the energy function to be minimized. This energy is based on the distance between the current CAD models $\mathcal{M}_i^\kappa(t)$ and their cropped point clouds PC_i^κ . Thus, at each step of the optimization loop, the CAD modeler updates the $\mathcal{M}_i^\kappa(t)$ according to the values $x_{j,k}(t)$. However, the possible assembly structure is not updated at this step, but after the ICP registration as discussed in Sect. 4.4.3. Once updated, each CAD model is then tessellated to get the triangle meshes $\mathcal{M}_{i\triangleright}^\kappa(t)$. Actually, in the proposed implementation, the tessellated models are directly the ones generated by the modeler for the visualization purposes, thus they generally contain few triangles with widely varying shapes and dimensions. Then, it is possible to compute the deviation between each $\mathcal{M}_{i\triangleright}^\kappa(t)$ and its PC_i^κ by means of computing the distances between a point cloud and a mesh (Sect. 4.4.4).

4.4.3 Registration with ICP and assemblies update

The parameters $x_{j,k}(t)$ controlling the evolution of the CAD models $\mathcal{M}_i^\kappa(t)$ during the optimization loop do not affect the position and orientation of the parts with respect to the cropped point clouds PC_i^κ to be fitted. Indeed, adding six additional parameters to control the position and orientation of each part (i.e. $6 \times N_m$ additional control parameters overall) would clearly reduce the performances of the SA algorithm. Thus, once the $\mathcal{M}_i^\kappa(t)$ have been updated, their position and orientation are modified using an ICP algorithm [27] that finds a best fit rigid body transformation between each $\mathcal{M}_{i\triangleright}^\kappa(t)$ and its PC_i^κ . Then, if it exists, the assembly structure $\mathcal{A}(t)$ is updated by the CAD modeler and the assembly constraints are satisfied.

4.4.4 Distance computation

Once the possible assembly structure $\mathcal{A}(t)$ has been updated, and once the $\mathcal{M}_i^\kappa(t)$ have been updated, tessellated and registered, the deviation between PC_i^κ and $\mathcal{M}_{i\triangleright}^\kappa(t)$ can be computed as follows:

$$d(PC_i^\kappa, \mathcal{M}_{i\triangleright}^\kappa(t)) = \sum_{\tau=1}^{n_i^\kappa} d^2(PC_i^\kappa[\tau], \mathcal{M}_{i\triangleright}^\kappa(t)), \quad (5)$$

where n_i^κ is the number of points in PC_i^κ , and $d(\text{point}, \text{mesh})$ is the distance function that returns the closest distance between a point and a mesh. In the proposed implementation, this evaluation is performed efficiently by CloudCompare called in batch mode. When considering several CAD models to be fitted simultaneously, the deviations are computed for each model \mathcal{M}^κ with $\kappa \in \{1, \dots, N_m\}$, and each

contribution is added up in the overall energy function to be minimized (Eq. 4).

4.5 End of the loops

The three loops operate in a nested manner. Once the optimal parameters values have been found in the related optimization loop, the algorithm goes back to the parameters loop if $j < j_{\max}$, otherwise it goes back to the segmentation loop and if $i < i_{\max}$ it starts a new segmentation. When all the loops are finished, the $\mathcal{M}_{i_{\max}}^k$ correspond to the fitted CAD models, and the $PC_{i_{\max}}^k$ to the by-part segmentations of the original point cloud.

This is illustrated on Fig. 2 which shows the evolution of the cropped point clouds PC_i during the segmentation loops, as well as the final fitted flange $\mathcal{M}_{i_{\max}}$ and associated segmentation $PC_{i_{\max}}$. The numerical results associated to this example are discussed in Sect. 5.6.

5 Results and discussion

This section aims at presenting some additional results to validate various fitting scenarios. It briefly introduces the technique used to generate as-scanned point clouds, as well as the procedure to tune the initial temperature of the SA algorithm. Four experimentations are then presented and discussed. The first example illustrates the global fitting of a L-like shape in an as-scanned point cloud. It allows the validation and benchmark of the proposed approach when considering more or less noisy data. The second example is used to validate the proposed fitting strategy on a real scanned point cloud. The third example illustrates how the fitting approach can be used to track the position and orientation of robot arms, and thus update the digital twin of a physical system in the scope of the Industry 4.0. The fourth example is more complex as it demonstrates how to update a CAD assembly model following a part-by-part fitting strategy, while also considering the simultaneous fitting of several parts as well as the possibility to handle assembly constraints during the optimization loops.

The core of the fitting algorithm has been implemented in MATLAB[®], which is able to call the built-in functions of SolidWorks[®] to perform the successive updates and ensure the consistency of the resulting B-Rep models during the optimization loops. The ICP algorithm is run in CloudCompare also called in batch mode.

5.1 As-scanned point cloud generation

To experiment and validate the proposed fitting strategy, it is mandatory to rely on ground truth data. Thus, the overall

fitting framework has been customized to allow bypassing the scanning of a real-life mechanical assembly (dashed lines of Fig. 1) and to enable the use of as-scanned virtually generated point clouds. As-scanned point clouds are automatically generated from CAD models. The resulting point clouds incorporate various realistic artifacts (e.g. non-uniform sampling, missing data, noise and outliers) that would appear if the corresponding real objects were digitalized with a real acquisition device. The details of the virtual scanning algorithm can be found in the paper of Montlahuc et al. [23]. Overall, the generation of an as-scanned point cloud follows a six-step sequential process: wrapping of the assembly model to produce a watertight triangle mesh, resampling to control the density, removal of hidden points to simulate the occlusion phenomenon, generation of misalignments to simulate multiple poses, insertion of noise and outliers. Each step is optional. The removal of points follows the efficient HPR algorithm of Katz et al. [28], and the insertion of noise is obtained, while moving points along the line of sight using a Gaussian random distribution. This is illustrated on Fig. 4.

Following this virtual scanning process, the final values p_k^F of all the parameters controlling the fitted CAD models and assembly structure can be compared to the original values p_k^D of the ground truth models as they appear in the DMU. As a consequence, for each parameter, the relative deviation δp_k and the absolute deviation Δp_k can be computed as follows:

$$\Delta p_k = \frac{|\delta p_k|}{|p_k^D|} = \frac{|p_k^D - p_k^F|}{|p_k^D|} \quad \forall k \in [1 \dots N_p] \quad (6)$$

with $N_p = \sum_{j=1}^{j_{\max}} N_{pj}$,

where N_p stands for the overall number of parameters possibly distributed in the parameters groups. The absolute deviation is computed at the end of each segmentation loop ($\forall i \in \{1, \dots, i_{\max}\}$) so as to appreciate the convergence step after step, whereas the relative deviation is computed

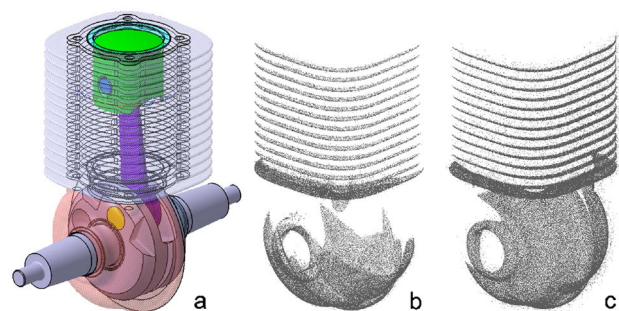


Fig. 4 Generation of as-scanned point clouds [23]: **a** CAD assembly model to be virtually scanned, **b** point cloud without noise, and with noise, **c** using different view points

on the final configuration only ($i = i_{\max}$) so as to assess the accuracy of the overall fitting process.

5.2 Tuning of the SA algorithm initial temperature

For an efficient search of the x_k optimal numerical values, the SA algorithm initial temperature T_0 needs to be tuned. In the proposed implementation, T_0 is initialized before entering the three nested loops and following a procedure similar to the one of Ben-Ameur [29]. More precisely, the initialization procedure looks for the optimal temperature T_0 that maximizes the decrease of the energy function in a measuring window of width W_{iter} , i.e. between iteration 1 and W_{iter} . This is performed while launching several times the SA algorithm with different T_0 . For each T_0 , the value of the energy function is tracked during the W_{iter} very first iterations of the SA algorithm, and the temperature which gives rise to the most important decrease of the energy function is selected. Actually, to smooth the effect of the stochastic behavior, the energy function is averaged while considering its values during the L_{iter} last iterations of the measuring window, i.e. between iterations $(W_{\text{iter}} - L_{\text{iter}})$ and W_{iter} . As a consequence, the initial temperature T_0 of the SA algorithm varies depending on the fitting scenarios.

In the following examples, selecting a measuring width of $W_{\text{iter}} = 25$ first iterations and considering the $L_{\text{iter}} = 5$ last iterations of this window to smooth the energy values have proved to be good trade-off between waiting a sufficient stabilization of the stochastic behavior, and avoiding too much iterations. Actually, these values must be compared to the 150 iterations usually required for the SA algorithm to converge, i.e. to reach a max number of iteration M_{iter} without any change of the energy function (up to a given accuracy ϵ_{iter}). Moreover, in the proposed implementation, the values of T_0 to be tested evolve between 5 and 50 with an increment of 5. Thus, ten simulations (stopped after W_{iter} iterations) are necessary to identify the best initial temperature for a given fitting example.

5.3 Global fitting of a symmetric L-like shape

The first example deals with the global fitting of a symmetric L-like shape controlled by five parameters ($N_a = 0$, $N_m = 1$ and $N_p = 5$): the lengths L and ℓ , the thickness e , the holes radius R and the size of the fillet r . This part has been rapidly sketched and parameterized in a pre-processing step. Three groups of parameters are defined according to the level of detail they correspond to (Fig. 5 and Table 1). This decomposition conforms to the classical CAD modeling strategy, which starts from structural and detail features, and ends up with skinning features. From the original CAD model, an as-scanned point cloud has been created using the HPR algorithm from 6 viewpoints. The point cloud is composed of about 500k points. At first, no noise has been added. As it is a global fitting, no segmentation is performed and the optimization loop is run on the three parameter groups. At each iteration, the ICP algorithm is run to control the position and orientation of the updated part with respect to the point cloud.

Following the initialization procedure detailed in Sect. 5.2, the initial temperature T_0 of the SA algorithm is set up to 10 for the L-like shape fitting example. This is the value of T_0 that maximizes the decrease of the energy function in the very first iterations. The algorithm stops when a max number of iteration ($M_{\text{iter}} = 50$) without any change (up to an accuracy $\epsilon_{\text{iter}} = 10^{-1}$) is reached. The algorithm runs for the successive segmentations (segmentation loop), and for the three groups of parameters (parameters loop). The graphics of Fig. 6 show the SA evolution curves for the three parameters groups \mathcal{G}_j of the last segmentation loop ($i = 3$). The group \mathcal{G}_1 is optimized first, then \mathcal{G}_2 , and \mathcal{G}_3 at last. The stochastic nature of the SA algorithm is clearly visible. Such an algorithm is interesting to solve this global optimization problem in a large search space. Of course, in scenarios where the final parameters values are to be rounded, the minimization process could stop earlier while acting on the value of ϵ_{iter} .

Figure 5 shows the fitted L-like shape and the numerical results are gathered together in Table 1. The average

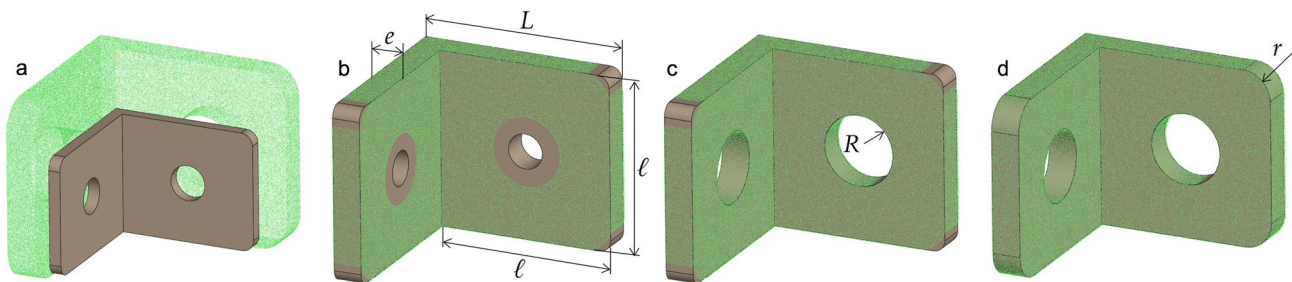


Fig. 5 Global fitting of a L-like shape following three optimization loops: **a** coarse pre-arrangement in the initial point cloud; **b** loop on \mathcal{G}_1 ; **c** loop on \mathcal{G}_2 ; **d** final fitted part after a loop on \mathcal{G}_3

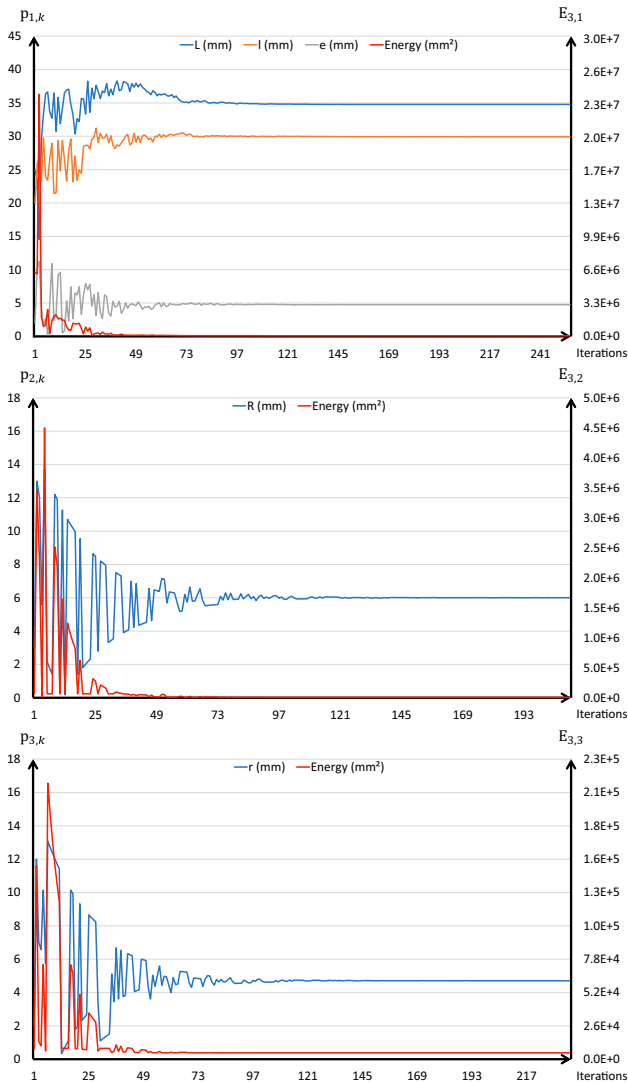


Fig. 6 Simulated annealing evolution curves for the three parameters loops ($j \in \{1, \dots, 3\}$) of the last segmentation step ($i = 3$) during the global fitting of a L-like shape: parameters values evolution (left) and energy evolution (right)

absolute deviation between the point cloud and the CAD model is about 0.1133 mm (min = 0.0 mm, max = 0.3907 mm, std = 0.0495 mm) and the average relative deviation is about 0.0034 (when using the average of the three dimensions of the oriented bounding box as a reference

distance), which is quite low. The relative and absolute deviations between the parameters final values p_k^F and the ones of the original part p_k^D are low. Actually, the largest deviations are for parameters e and r . This is due to the fact that those parameters are controlling relatively small features whose size variations have a low impact on the energy function to be minimized. This effect is more deeply analyzed in Sect. 5.4 where alternative solutions are sketched. Using this part-by-part fitting strategy, the final L-like shape better fits the point cloud than if a traditional patch-by-patch reconstruction process would have been followed. Indeed, the L-like shape is fitted globally and the final solution does not depend on a reference surface which would have been fitted individually at the beginning of the manual reconstruction process. Moreover, internal constraints (e.g. symmetry, perpendicularity, holes centered on the faces) are directly managed by the CAD modeler, which handles the successive updates and maintains the CAD model consistency step after step. Finally, if the parameters would not have been distributed in the three groups, the results would have been less good. Indeed, when considering the five parameters at the same level, the average absolute deviation is about 0.2636 mm (min = 0.0 mm, max = 2.2474 mm, std = 0.4264 mm) and the average relative deviation is about 0.0080, and both values are higher than when considering the parameters within the three groups. This validates the proposed decomposition strategy.

Another experimentation has been performed to study the impact of the presence of noise on the fitting results. To this aim, a noise has been added to the whole point cloud using a Gaussian noise random distribution controlled by an amplitude factor. Results of the fitting process with two levels of noise (50 μ m and 100 μ m) are listed in Table 2. One can clearly observe that as the noise increases, the relative deviation also increases coherently. This effect is more significant for small features. Obviously, the levels of noise which have been applied are far greater than what can appear on a well-calibrated acquisition device used in normal conditions. Thus, the proposed approach is relatively stable to noise, except for small detail features which can hardly be well fitted in the presence of too much noise. But in this case, following a more conventional manual reconstruction process

Table 1 Results for the global fitting of a L-like shape (without noise and with an initial temperature of the SA algorithm $T_0 = 10$)

Groups	p_k	p_k^0 (mm)	p_k^D (mm)	p_k^F (mm)	δp_k (mm)	Δp_k
\mathcal{G}_1	L	25	35	34.7800	0.2200	0.0063
	ℓ	20	30	29.9210	0.0790	0.0026
	e	2	5	4.7514	0.2486	0.0497
\mathcal{G}_2	R	3	6	6.0045	-0.0045	0.0008
\mathcal{G}_3	r	2	5	4.7061	0.2939	0.0588

would lead to the same difficulties. Anyhow, depending on the adopted fitting scenario, the obtained parameter values may also be rounded at the end.

5.4 Global fitting of a sonotrode to a real scanned point cloud

This second example studies the behavior of the fitting algorithm when dealing with real scanned data. This time, a ROMER Absolute Arm 7520 SI (7 axis, 2m acquisition volume, absolute encoders, RSI laser sensor 30000pts/s) has been used to scan a Sonotrode (Fig. 7a) and get a point cloud of about 257k points (Fig. 7b). The resulting point cloud incorporates noise and artifacts. The CAD model of the Sonotrode to be fitted is controlled by ten parameters split in three groups ($N_a = 0$, $N_m = 1$ and $N_p = 10$): structural features (\mathcal{G}_1), geometric features (\mathcal{G}_2) and skinning features (\mathcal{G}_3). Following the initialization procedure of Sect. 5.2, the initial temperature T_0 of the SA algorithm is set up to ten for the Sonotrode example, and the other control parameters are the same as for the previous example ($M_{\text{iter}} = 50$ and $\epsilon_{\text{iter}} = 10^{-1}$). The fitting process starts by the pre-arrangement of the CAD model with respect to the reference point cloud (Fig. 7b). This initializes the fitting algorithm, which then starts by optimization the structural parameters of \mathcal{G}_1 (Fig. 7c). For this loop, the holes D_3 and D_4 have been deactivated from the building tree. Doing this way, the fitting of D_1 is not affected by the internal points related to the holes, and the corresponding areas of the point cloud can be cropped through the segmentation loop. The parameters of

\mathcal{G}_2 are then optimized (Fig. 7d). The axes of the holes are constrained using built-in constraints (i.e. perpendicularity, coaxiality and circular repetition) of the CAD modeler in charge of the updates. Due to the occlusion phenomenon, there are too few points to accurately identify the depth of the holes, which, therefore, has not been considered as a parameter to be optimized. As part of the last optimization loop, the parameters of \mathcal{G}_3 are then optimized to get the final fitted CAD model of the Sonotrode (Fig. 7e). The numerical results are synthesized in Table 3.

This fitting problem is more complex and the difficulties are threefold: (i) the number of variables is greater than in the previous case, resulting in more iterations; (ii) some features are quite small when compared to the size of the object, and their contribution within the energy function becomes negligible, thus causing greater deviations of the corresponding parameters

Table 2 Evolution of the relative deviation Δp_k with respect to the amplitude of the inserted noise (0 μm , 50 μm , 100 μm) for the global fitting of the L-like shape

Groups	p_k	Relative deviation Δp_k		
		0 μm	50 μm	100 μm
\mathcal{G}_1	L	0.0063	0.0068	0.0187
	ℓ	0.0026	0.0000	0.0099
	e	0.0497	0.0553	0.1315
\mathcal{G}_2	R	0.0008	0.0018	0.0192
\mathcal{G}_3	r	0.0588	0.0327	0.1332

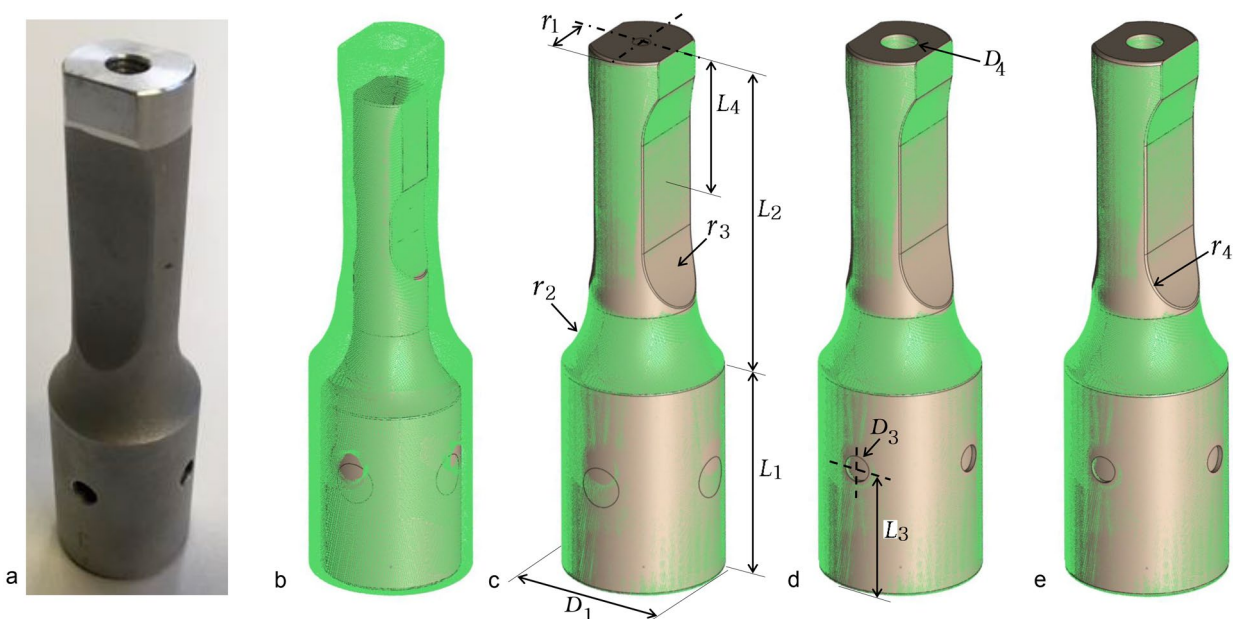


Fig. 7 Global fitting of a Sonotrode to a real scanned point cloud following 3 optimization loops: **a** scanned Sonotrode; **b** coarse pre-arrangement in the initial point cloud; **c** loop on \mathcal{G}_1 ; **d** loop on \mathcal{G}_2 ; **e** final fitted part after a loop on \mathcal{G}_3

Table 3 Results for the global fitting of a Sonotrode to a real point cloud obtained by a laser scanner

Groups	p_k	p_k^0 (mm)	p_k^D (mm)	p_k^F (mm)	δp_k (mm)	Δp_k
\mathcal{G}_1	D_1	30	35	34.9872	0.0128	0.0004
	L_1	40	47	47.6183	-0.6183	0.0132
	L_2	70	75	74.0385	0.9615	0.0128
	r_1	7	11.5	11.4656	0.0344	0.0030
	r_2	22	24	21.1645	2.8355	0.1181
	r_3	12	20	20.0028	-0.0028	0.0001
\mathcal{G}_2	D_3	9	6	6.6348	-0.6348	0.1058
	D_4	4	8	7.9101	0.0899	0.0112
	L_3	27	30	29.7042	0.2958	0.0099
\mathcal{G}_3	r_4	0.5	1.5	0.8376	0.6624	0.4416

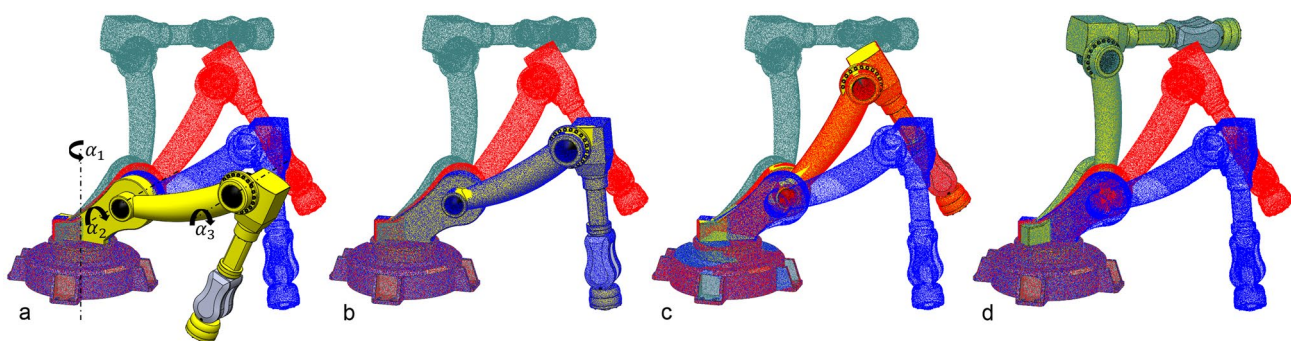
values (e.g. parameter r_4 in \mathcal{G}_3); (iii) due to the occlusion phenomenon, the internal holes cannot be well captured by the acquisition device, resulting in difficulties to get a good fitting of the internal holes' parameters (i.e. parameters in \mathcal{G}_2 and depth of the holes). Despite those difficulties, the average absolute deviation between the point cloud and the CAD model of the Sonotrode is about 0.0622 mm (min = 0.0 mm, max = 5.8095 mm, std = 0.3536 mm) and the average relative deviation is about 0.0010, which is quite low. The larger values of min and max can directly be ascribed to the issue (iii).

This example shows that the size of the features clearly influences the quality of the fitting. This is directly linked to the contribution of those features within the overall energy function to be minimized. This issue could be overcome while considering a particular weighting strategy for each feature of the CAD model to be fitted. As a consequence, in a pre-processing step, the parameterized CAD model could be used to compute the sensitivity of each parameter p_k to shape variations, and then infer the proper distribution of the weights. This is further discussed in the conclusion.

5.5 Local fitting of robot arms

The third example illustrates how the fitting approach can be used to track successive moves of robot arms, and thus update the digital twin of a physical system (Fig. 8). The robot is supposed to be stopped during the update of its digital twin. Such a possibility is particularly interesting to maintain the coherence between the physical system and its digital twin in the scope of the Industry 4.0 [1].

The DMU of the considered robot is composed of three arms and one fixed support ($N_m = 3$). The orientation of the arms with respect to the others is parameterized by three angles α_i (with $i \in [1 \dots 3]$) gathered in a single group \mathcal{G}_1 (Fig. 8a). The shape of the arms is not considered in this test case ($N_p = 0$), which focuses on the global fitting of the assembly structure ($N_a = 3$). Here, the objective is to be able to retrieve the values of the arms' rotation parameters after three moves of the robot. To this aim, three as-scanned point clouds have been created using the HPR algorithm from six viewpoints, thus generating a bunch of 614 k/520 k/410 k points for, respectively, the first, second and third move. Following the initialization procedure of Sect. 5.2, the initial temperature T_0 of the SA algorithm is set up to 25 for the robot arms example, and the other control parameters

**Fig. 8** Fitting of robot arms constrained with assembly constraints and parameterized by 3 rotation angles: **a** initial configuration; **b–d** final configurations fitting respectively the first, second and third robot moves

are the same as for the previous example ($M_{\text{iter}} = 50$ and $\epsilon_{\text{iter}} = 10^{-1}$). Starting from an initial configuration, the fitting algorithm is sequentially called three times so that the robot arms successively fit the three virtually generated point clouds (Fig. 8b–d).

The numerical results are provided in Table 4. It clearly shows that the parameters' values obtained after a fitting serves as new initial values for the next fitting. As in the previous test cases, the relative deviations of the control parameters are very low. In the worst case, i.e. for the move that gives the largest deviation, the average absolute deviation between the point cloud and the CAD models is about 0.0068 mm (min = 0.0 mm, max = 5.5210 mm, std = 0.1071 mm) and the average relative deviation is about 0.00001, which is also quite low. Actually, when only dealing with assembly parameters only, the fitting process is very efficient as it does not require CAD models updates. This demonstrates the pertinence of the proposed technique to tackle such a fitting scenario in the scope of the Industry 4.0. This is a first step towards the accurate tracking of robots evolving in complex industrial environments.

5.6 Local fitting of multiple parts in a valve assembly

The last example illustrates how the fitting strategy can be used to update the CAD assembly model of a valve. Here, it is assumed that the CAD models already exist and need to be updated. To be able to validate the approach, from the original DMU made of 40 assembled parts, an as-scanned point cloud has been created using the HPR algorithm from 10 viewpoints. The resulting point cloud is composed of 1204 k points. No noise has been added. The reconstruction process follows a part-by-part fitting strategy where several parts can be fitted simultaneously, while also satisfying assembly constraints. Each part is defined by several control parameters, whose values are to be optimized (Table 5). In this scenario, the parts with the biggest extent are fitted first so as to get a good initial fitting, which will then serve as a reference for the fitting of the other parts. However, as it will be explained,

during the successive fittings, parameters of previously fitted parts can still be reconsidered as variables for the upcoming fitting steps. Indeed, considering more points step after step can improve the fitting accuracy of previously fitted parts. Of course, if the final objective is to control possible misalignments between assembled parts, assembly constraints should not be specified so as to let the algorithm capturing the as-is assembly configuration, but this is not the considered scenario here. The SA control parameters are tuned as for the previous examples ($M_{\text{iter}} = 50$ and $\epsilon_{\text{iter}} = 10^{-1}$), except for the initial temperature T_0 that is tuned differently for each of the parts to be fitted.

At first, after a pre-arrangement of the two flanges in the point cloud, the initial temperature T_0 of the SA algorithm is set up to 15 while following the initialization procedure of Sect. 5.2. The two flanges are fitted simultaneously using eight parameters to control the shape of the flanges and one parameter to control the distance between the two flanges ($N_m = 2$, $N_p = 8$ and $N_a = 1$). Additional assembly constraints are also used and directly handled by the CAD modeler in charge of the successive updates (Fig. 9.a₁ and a₂): the axes of the cylindrical faces of radius r_1 have to be coincident (orange color), so do the axes of the through holes of radius r_5 . Overall, the deviations between the final parameters values and the ones of the original part are low. Actually, the largest deviations are for parameters r_5 and r_6 . This is due to the fact that those radii are rather small and very few points can contribute to the fitting. Anyhow, a more conventional reverse engineering process would lead to the same issue. Since the two flanges are fitted simultaneously, the resulting deviations are smaller than if the flanges would have been fitted sequentially. This is due to the fact that the number of points and their distribution in the 3D space make this configuration more stable than when fitting the flanges one after the other.

Once the two flanges fitted, the update process keeps on going with the simultaneous fitting of two identical screws defined by four control parameters (Fig. 9.b₁ and b₂). For this fitting step, the parameter r_4 of the flanges is reconsidered as a variable that can be further optimized during the fitting of

Table 4 Results obtained when fitting three successive moves of robot arms parameterized by three rotation angles

Groups	p_k	p_k^0 (mm)	p_k^D (mm)	p_k^F (mm)	δp_k (mm)	Δp_k
\mathcal{G}_1 Move-1	α_1	75	90	89.9990	0.0010	0.0000
	α_2	15	30	29.9946	0.0054	0.0002
	α_3	65	95	94.9968	0.0032	0.0000
\mathcal{G}_1 Move-2	α_1	89.9990	115	114.9987	0.0013	0.0000
	α_2	29.9946	55	54.9954	0.0046	0.0001
	α_3	94.9968	120	119.9976	0.0024	0.0000
\mathcal{G}_1 Move-3	α_1	114.9987	140	139.9985	0.0015	0.0000
	α_2	54.9954	90	89.9935	0.0065	0.0001
	α_3	119.9976	170	170.0019	-0.0019	0.0000

the screws. Thus, the SA algorithm tries to find the optimal values of five parameters (four parameters of the screws, and one parameter of the flanges). The pre-arrangement is performed while using the axes of the previously fitted flanges. Each screw is constrained to have its axis coincident with the axis of the flange's through hole (orange and purple colors), and the front planar face of the flange should be coincident with the contact face of the screw head (blue color). Following the initialization procedure of Sect. 5.2, the initial temperature T_0 of the SA algorithm is set up to 10 for the screws fitting. Here again, as the two screws are fitted simultaneously, the final parameters deviations are better controlled. Similarly, the two nuts are then fitted based on the previously fitted screws and flanges (Fig. 9.c₁ and c₂). Each nut is parameterized by two control parameters and it is constrained to have its axis coincident with the axis of the screw (orange and purple), and the bottom face of the nut should be coincident with the planar face of the flange (blue color). For this fitting step, the parameter r_4 of the two flanges and the diameter bd of the screws are reconsidered as variables that can be further optimized during the fitting of the nuts. Thus, the SA algorithm tries to find the optimal values of four parameters (two parameters of the nuts, one parameter of the flange and 1 parameter of the screw). In addition, the diameter bd of both the screws and nuts are linked to the radius r_5 of the fitted flanges. Here, the initial temperature T_0 of the SA algorithm is set up to 5 for the nuts fitting.

Following the part-by-part fitting strategy, the central part is then fitted while optimizing the values of five control parameters (Fig. 9.d₁ and d₂). Using the same coloring strategy, the axes of the two parts are constrained to be coincident, and the lateral contact faces of the central part are constrained to be coincident with the back faces of the two flanges. Here, the initial temperature is set up to 5. Then, the top plate is fitted to the point cloud. It is parameterized by five control parameters and aligned with the central part using coincidence constraints on both the axes and contact faces (Fig. 9.e₁ and e₂). The initial temperature is set up to 10 for the top plate. For this fitting step, the parameter cpd_1 of the central part is reconsidered as a variable to be further optimized. Thus, the SA algorithm tries to find the values of six parameters (five parameters of the top plate, and one parameter of the central part). Finally, the bottom plate is fitted to the point cloud. It is parameterized by four control parameters and aligned with the central part with which it is also in contact (Fig. 9.f₁ and f₂). For this fitting step, the parameter cpd_2 of the central part is reconsidered as a variable to be further optimized. Thus, the SA algorithm tries to find the values of five control parameters (4 parameters of the bottom plate, and one parameter of the central part). Here, the initial temperature is set up to 5 for the bottom plate.

This example clearly shows the potential of the proposed part-by-part fitting strategy to update an entire CAD assembly model so that it fits the point cloud of a digitized assembly. The method is also able to segment the point cloud of the digitized valve so as to highlight the different parts (Fig. 9.g₁ to g₄). The relative deviations of the control parameters are quite low (Table 5), except for the radius of some fillets and for small features for which larger deviations can be ascribed to a lack of data in the corresponding areas of the point cloud. This has been discussed for the previous experiments. Here again, depending on the adopted scenario, the obtained parameter values may be rounded at the end. Of course, being the screws and nuts standard elements, the obtained parameters values can serve as a reference to select in a catalog the closest available ones. Finally, Table 6 gathered together the absolute and relative errors measured between the point cloud and each fitted part. Those errors are rather small when compared to the size of the parts.

6 Conclusion and future works

This paper has introduced a new framework able to fit simultaneously several parameterized CAD models in the point cloud of a digitized assembly. CAD models to be fitted can also be constrained with assembly constraints. Both the consistency of the CAD models and the constraints between them are managed by the CAD modeler which acts every time the parameters are modified. The resulting CAD models can directly be used and edited in the stages of the PDP. When considering local fitting, the initial point cloud is also segmented at the end of the process. The proposed approach is very promising when considering the need to maintain the coherence between a physical system and its digital twin in the scope of the Industry 4.0. It bypasses the traditional tedious and time-consuming patch-by-patch reverse engineering process. It makes use of a simulated annealing algorithm to find the optimal values in a large search space. The framework has been designed around three nested loops which act at different levels: the segmentation loop that isolates step after step the subset of the original point cloud to which the CAD models have to be fitted; the parameters loop which splits the optimization problem according to three categories of parameters, similar to what is traditionally used when modeling CAD parts and assemblies in a CAD environment; and the optimization loop which runs the simulated annealing algorithm on the subproblems. The method has proved its efficiency for global and local fitting. It has been tested on several configurations, for which as-scanned point clouds have been generated following an ad hoc virtual scanning approach. Thus, it has been possible to measure the deviations between the parameters of the fitted parts, and the ones of the original parts as they appear in the DMU. The

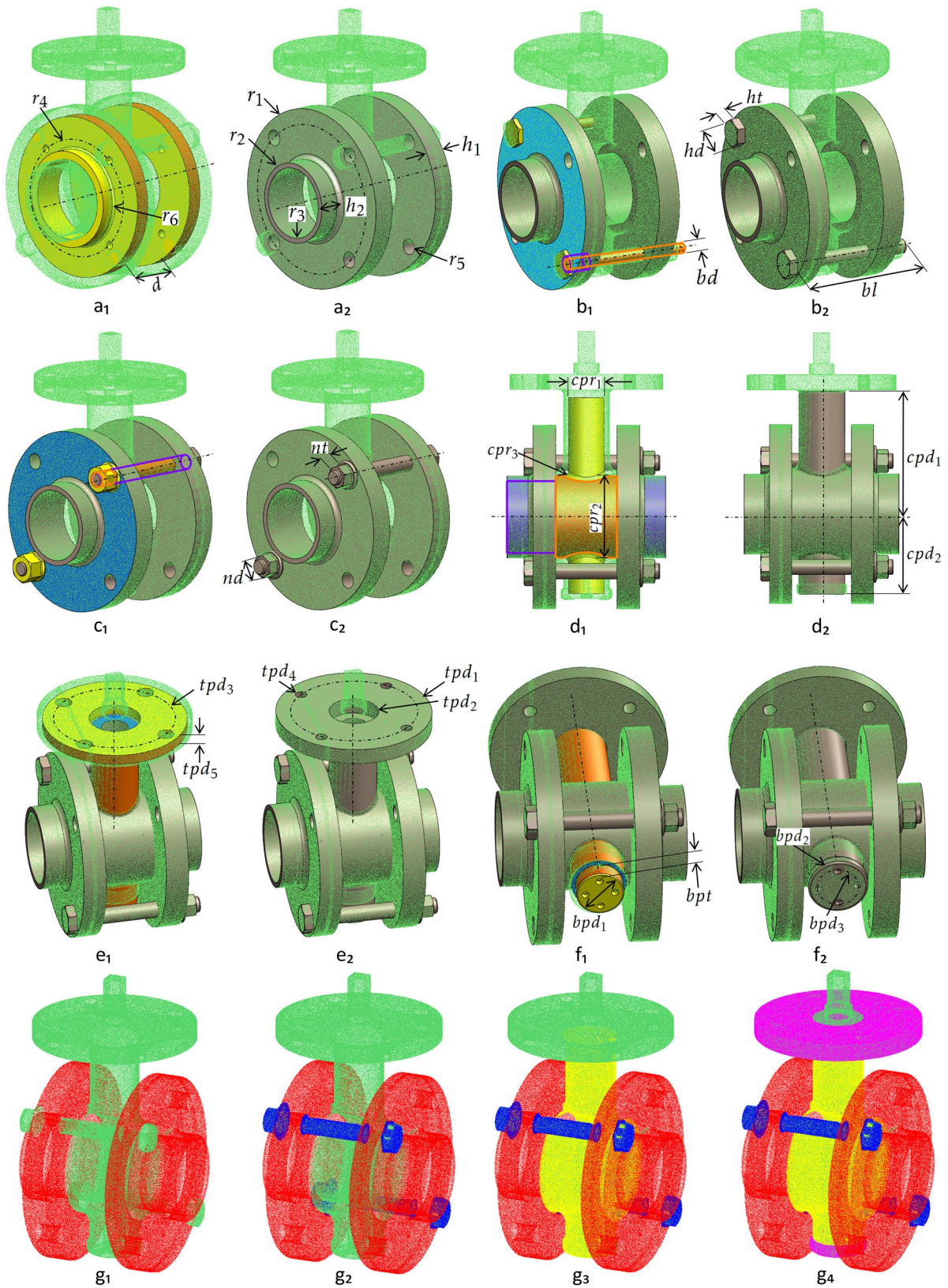


Fig. 9 Successive fittings of multiple CAD models in the point cloud of a digitized valve assembly: (a₁) coarse pre-arrangement of two flanges in the initial point cloud; (a₂) final fitted flanges; (b₁) pre-arrangement of two screws; (b₂) final fitted screws; (c₁) pre-arrangement of two nuts; (c₂) final fitted nuts; (d₁) pre-arrangement of the central part; (d₂) final fitted central part; (e₁) pre-arrangement of the top plate; (e₂) final fitted top plate; (f₁) pre-arrangement of the bottom plate; (f₂) final fitted bottom plate; (g₁ to g₄) successive segmentations of PC₀

influence of the noise and of the parameters grouping on the final results have also been analyzed. The experimentations have been performed on mechanical parts assemblies, but the method could be extended to allow the more general local fitting of parameterized objects in virtual environments. A first attempt to define a tracking approach able to maintain the coherence between a physical system and its digital twin has proved to be promising, and it already gives good results on a simple 3-axes robot.

The various experimentations allowed to demonstrate several interesting features of the approach, as well as ways of improvement. Among others, it is clearly demonstrated that for the local fitting, a sufficient amount of information has to be accessible. This is of course very similar to the traditional manual reconstruction process, and the proposed approach cannot capture the invisible. This is notably true to fit small features as well as to capture interfaces between parts. Thus, few disassembly steps might be required before scanning. The exploitation of RANSAC-based, or even manually, extracted primitives to be used as constraints of the fitting problem could help solving this issue. The integration of this fitting technique as part of a reverse engineering framework might also help speeding the entire reconstruction process in case no parameterized model exists. However, this requires some adaption so as to be able to work with parameters of 2D sketches and 3D features, and thus to allow reconstructing a full CAD model step after step

Table 5 Results for the local fitting of multiple parts in a valve assembly



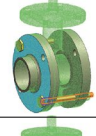

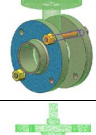

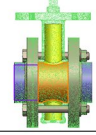





Pre-arranged parts(s)	Fitted part(s)	Groups	p_k	Initial values p_k^I (mm)	Target values p_k^D (mm)	Final values p_k^F (mm)			Absolute dev. δp_k (mm)			Relative dev. Δp_k
						$i=1$	$i=2$	$i=3$	$i=1$	$i=2$	$i=3$	$i=3$
Flanges												
		\mathcal{G}_1	r_1	80	90	90.0697	90.0697	90.171	-0.0697	-0.0697	-0.171	0.0019
			r_2	38	40	39.8267	39.8267	40.1883	0.1733	0.1733	-0.1883	0.0047
			r_3	30	35	34.4396	34.4396	34.9131	0.5604	0.5604	0.0869	0.0025
			h_1	15	22	23.8543	23.8543	22.2166	-1.8543	-1.8543	-0.2166	0.0098
			h_2	15	25	24.934	24.934	24.9712	0.066	0.066	0.0288	0.0012
			d	70	60	57.3323	57.3323	59.7876	2.6677	2.6677	0.2124	0.0035
		\mathcal{G}_2	r_4	60	75	74.9759	74.9759	74.9759	0.0241	0.0241	0.0241	0.0003
			r_5	5	8	7.4503	7.4503	7.4503	0.5497	0.5497	0.5497	0.0687
		\mathcal{G}_3	r_6	2	5	0.3446	3.7187	4.5439	4.6554	1.2813	0.4561	0.0912
		Screws										
		\mathcal{G}_1	hd	21	24	24.7793	24.7793	24.3001	-0.7793	-0.3001	-0.3001	0.0125
			ht	5	8	8.9433	8.9433	8.895	-0.9433	-0.9433	-0.895	0.1119
			bl	130	121	139.959	139.959	120.1898	-18.959	-18.959	0.8102	0.0067
			bd	10	14	14.3687	14.3687	14.2536	-0.3687	-0.3687	-0.2536	0.0181
			r_4	74.9759	75	74.9225	74.9225	74.9684	0.0775	0.0775	0.0316	0.0004
Nuts												
		\mathcal{G}_1	r_4	74.9684	75	74.9868	74.9807	75.0393	0.0132	0.0193	-0.0393	0.0005
			bd	14.2536	14	14.4136	14.4408	14.7502	-0.4136	-0.4408	-0.7502	0.0536
			nd	25	21	21.052	20.7961	20.8564	-0.052	0.2039	0.1436	0.0068
			nt	15	12	14.0638	14.5033	14.807	-2.0638	-2.5033	-2.807	0.2339
Central part												
		\mathcal{G}_1	cpd_1	117	125	118.052	123.99	123.8636	6.948	1.01	1.1364	0.0091
			cpd_2	75	70	81.1333	79.7596	80.1172	-11.1333	-9.7596	-10.1172	0.1445
			cpr_1	35	45	45.0812	46.0554	45.8565	-0.0812	-1.0554	-0.8565	0.0190
			cpr_2	80	90	87.2774	89.4893	89.7125	2.7226	0.5107	0.2875	0.0032
			cpr_3	5	10	0.5925	0.0008	5.3119	9.4075	9.9992	4.6881	0.4688
Top plate												
		\mathcal{G}_1	tpd_1	140	150	145.1966	150.2271	150.0571	4.8034	-0.2271	-0.0571	0.00038
			tpd_2	50	45	40.8289	38.9789	40.4789	4.1711	6.0211	4.5211	0.10047
			tpd_3	115	120	115.4108	115.746	118.9028	4.5892	4.254	1.0972	0.00914
			tpd_4	16	12	6.4896	9.3327	11.4995	5.5104	2.6673	0.5005	0.04171
			tpd_5	10	15	15.0677	15.1752	15.1411	-0.0677	-0.1752	-0.1411	0.00941
cpd_1	123.8636	125	124.6546	124.7564	124.807	0.3454	0.2436	0.193	0.00154			
Bottom plate												
		\mathcal{G}_1	bpd_1	40	50	47.9315	48.3134	49.0784	2.0685	1.6866	0.9216	0.0184
			bpd_2	1	6	0.046	0.1388	4.1624	5.954	5.8612	1.8376	0.3063
			bpd_3	25	35	28.0237	30.7956	33.1279	6.9763	4.2044	1.8721	0.0535
			bpt	15	10	17.2641	17.0953	18.8609	-7.2641	-7.0953	-8.8609	0.8861
			cpd_2	67	70	62.9326	63.1342	61.35	7.0674	6.8658	8.65	0.1236

Table 6 Absolute and relative errors measured between the point cloud and each locally fitted part of the valve assembly

Parts	Absolute error (mm)					Relative error				
	Min	Max	Median	Mean	Dev.Std.	Min	Max	Median	Mean	Dev.Std.
Flanges	-1.9720	1.9065	-0.0067	-0.0118	0.2056	0.0115	0.0111	0.0000	0.0001	0.0012
Screws	-1.0178	1.2258	0.0781	0.0290	0.4098	0.0078	0.0093	0.0006	0.0002	0.0031
Nuts	-2.0938	2.1350	0.0741	0.0582	0.4286	0.0156	0.0159	0.0006	0.0004	0.0032
Central Part	-2.1930	2.1742	0.2658	0.3320	0.5730	0.0186	0.0185	0.0023	0.0028	0.0049
Top plate	-1.0940	2.0927	-0.0030	0.0099	0.2063	0.0104	0.0199	0.0000	0.0001	0.0020
Bottom plate	-2.2872	3.7552	-0.0526	0.0868	0.7390	0.0586	0.0963	0.0013	0.0022	0.0189

following a feature-by-feature reconstruction strategy. This is part of future works.

The proposed approach is modular and each module can still be improved separately. As already explained, the energy function could be weighed to better reflect the influence of the different parameters on the shape variations. The weights could rely on a sensitivity analysis run at the beginning of the fitting process. Moreover, the current implementation is yet at the level of the prototype and it does not fully allow for real-time fitting. This is mainly due to the modularity of the implementation that calls external modules that need to exchange with the optimization kernel. Thus, a better integration of the different modules can drastically reduce the time spent to read/write/update the data in the various data structures of the different modules. In addition, in the current version, the thresholds used to tune the accuracy of the final results have been set up to quite low values so as to get very good fitting results. Of course, depending on the scenarios and parts to be fitted, a good trade-off between accuracy and time has to be found. Thus, the thresholds could be increased and the final results rounded. This is particularly true for instance for the standard parts that will anyhow be selected in a catalog and defined with predefined parameters values (e.g. screws, nuts). Parallelization can also be a good mean to speed up the overall process, while considering for instance that the parameters loops are executed step after step while letting the user free to interact and continue his/her work on the already fitted parts.

Finally, the use of parameterized CAD models as priors is a promising idea to avoid learning the parameters from the data as there are explicitly encoded in the CAD models. More generally, global regularity priors (e.g. symmetries, repetitions, relationships) are directly encoded in the CAD models and can thus be exploited when solving the fitting problem. Clearly, more domain-dependent knowledge still remain to be exploited to better support advanced treatments of geometric models. This could also help better solving the tracking of systems evolving in complex environments.

References

- Lu Y (2017) Industry 4.0: a survey on technologies, applications and open research issues. *J Ind Inf Integr* 6:1–10
- Bagci E (2009) Reverse engineering applications for recovery of broken or worn parts and re-manufacturing. *Adv Eng Softw* 40(6):407–418
- Falcidieno B, Giannini F, Léon J-C, Pernot J-P (2014) Processing free form objects within a product development process framework. In: Michopoulos JG, Paredis CJJ, Rosen DW, and Vance JM (eds) *Advances in Computers and Information in Engineering Research, Vol 1*. in ASME-Press, pp 317–344
- Berger M, Tagliasacchi A, Seversky LM, Alliez P, Guennebaud G, Levine JA, Sharf A, Silva CT (2016) A survey of surface reconstruction from point clouds. *Comput Gr Forum* 36(1):301–329
- Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24(6):381–395
- Schnabel R, Wahl R, Klein R (2007) Efficient ransac for point-cloud shape detection. *Comput Gr Forum* 26(2):214–226
- Schnabel R, Degener P, Klein R (2009) Completion and reconstruction with primitive shapes. *Comput Gr Forum* 28(2):503–512
- Bey A, Chaine R, Marc R, Thibault G (2012) Effective shapes generation for bayesian cad model reconstruction. In: *Proceedings of the 5th eurographics conference on 3D object retrieval*, pp 63–66
- Lari Z, Habib A (2014) An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. *ISPRS J Photogramm Remote Sens* 93:192–212
- Attene M, Falcidieno B, Spagnuolo M (2006) Hierarchical mesh segmentation based on fitting primitives. *Vis Comput* 22(3):181–193
- Mitra N, Wand M, Zhang HR, Cohen-Or D, Kim V, Huang Q-X (2013) Structure-aware shape processing. In: *SIGGRAPH Asia 2013 courses*, pp 1:1–1:20
- Li Y, Wu X, Chrysathou Y, Sharf A, Cohen-Or D, Mitra NJ (2011) Globfit: consistently fitting primitives by discovering global relations. *ACM Trans Gr* 30(4):52:1–52:12
- Monszpart A, Mellado N, Brostow GJ, Mitra NJ (2015) Rapter: rebuilding man-made scenes with regular arrangements of planes. *ACM Trans Gr* 34(4):103:1–103:12
- Nan L, Xie K, Sharf A (2012) A search-classify approach for cluttered indoor scene understanding. *ACM Trans Gr* 31(6):137:1–137:10
- Ip CY, Gupta SK (2007) Retrieving matching cad models by using partial 3d point clouds. *Comput Aid Des Appl* 4(5):629–638
- Gelfand N, Mitra NJ, Guibas LJ, Pottmann H (2005) Robust global registration. *Symp Geom Process* 2(3):197–206
- Rabbani T, Van Den Heuvel F (2004) Methods for fitting csg models to point clouds and their comparison. In: *Proceedings of*

- the 7th IASTED international conference on computer graphics and imaging, Kauai, HI, USA 1719, pp 279–284
18. Buonamici F, Carfagni M, Furferi R, Governi L, Lapini A, Volpe Y (2018) Reverse engineering of mechanical parts: a template-based approach. *J Comput Des Eng* 5(2):145–159
 19. Wang J, Gu D, Yu Z, Tan C, Zhou L (2012) A framework for 3d model reconstruction in reverse engineering. *Comput Ind Eng* 63(4):1189–1200
 20. Stark R, Grosser H, Müller P (2013) Product analysis automation for digital mro based on intelligent 3d data acquisition. *CIRP Ann Manuf Technol* 62(1):123–126
 21. Bénéière R, Subsol G, Gesquière G, Le Breton F, Puech W (2013) A comprehensive process of reverse engineering from 3d meshes to cad models. *Comput Aid Des* 45(11):1382–1393
 22. Xu M, Li M, Xu W, Deng Z, Yang Y, Zhou K (2016) Interactive mechanism modeling from multi-view images. *ACM Trans Gr* 35(6):236:1–236:13
 23. Montlahuc J, Shah GA, Polette A, Pernot J-P (2019) As-scanned point clouds generation for virtual reverse engineering of cad assembly models. *Comput Aid Des Appl* 16(6):1171–1182
 24. Lupinetti K, Pernot J-P, Monti M, Giannini F (2019) Content-based cad assembly model retrieval: survey and future challenges. *Comput Aid Des* 113:62–81
 25. Gouaty G, Fang L, Michelucci D, Daniel M, Pernot J-P, Raffin R, Lanquetin S, Neveu M (2016) Variational geometric modeling with black box constraints and dags. *Comput Aid Des* 75:1–12
 26. Pernot J-P, Michelucci D, Daniel M, Fougou S (2019) Towards a better integration of modelers and black box constraint solvers within the product design process. *Ann Math Artif Intell* 85(2):147–173
 27. Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. *IEEE Trans Pattern Anal Mach Intell* 14(2):239–256
 28. Katz S, Tal A, Basri R (2007) Direct visibility of point sets. *ACM Trans Gr* 26(3):24:1–24:11
 29. Ben-Ameur W (2004) Computing the initial temperature of simulated annealing. *Comput Optim Appl* 29:369–385

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

2.2 Conclusion

This chapter introduced a new fitting technique for the reconstruction of editable parametric CAD models for parts and assemblies that can later be used in the stages of Product Development Processes. The proposed method bypasses the time-consuming traditional Reverse Engineering which normally ends up with the reconstruction of dead models that cannot be modified. The method is modular where different modules perform the different tasks and are linked together. Each module used for the fitting can further be improved. Through the selected examples, the proposed Simulated Annealing-based fitting technique has also shown very promising results when considering the need to maintain the coherence between the physical system and its digital twin in the scope of Industry 4.0. The method has proven its efficiency on the local and global fitting of CAD models. The way the parameters of SA can be tuned for different RE needs is improved with an approach based on the design of experiments (DoEs) and similarity assessment, as introduced in Chapter 3. The proposed method works on part-by-part reconstruction technique, so new features need to be added as: (1) the possibility to fit 2D sketches; this is important for RE starting from scratch; (2) better local identification of the parts and detection of the interfaces using a specific filter technique; (3) integration in a new RE framework. This will be defined in the Chapter 4.

CHAPTER

3

Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications

Contents

3.1	Introduction	54
3.2	Conclusion	68

3.1 Introduction

The third chapter focuses on the optimization of the proposed Simulated Annealing-based reverse engineering method (highlighted in green in Fig. 1). Due to the stochastic behavior of Simulated Annealing, it is not expected that it will always produce the same results. It is well known that a good parameter selection has a significant effect on the performance of the algorithm. An attempt has been made in this chapter to identify the optimal parameters of the Simulated Annealing algorithm to produce desirable fitting results with better quality reconstructed models having minimum deviations between the CAD models and the reference point clouds. The proposed technique integrates the sensitivity analysis of the CAD model's parameters to better understand the impact of Simulated Annealing on the parameters of the CAD models during the optimization process. This chapter also details a new framework that uses the Design of Experiment (DoEs) technique to create a small database of CAD models stored with their best-identified configurations for Simulated Annealing. Later, if a new CAD template is to be reconstructed with the SA-based fitting technique, a case-based approach can be applied for the setting of the SA parameters by reusing the values stored for the most similar CAD model in the database.

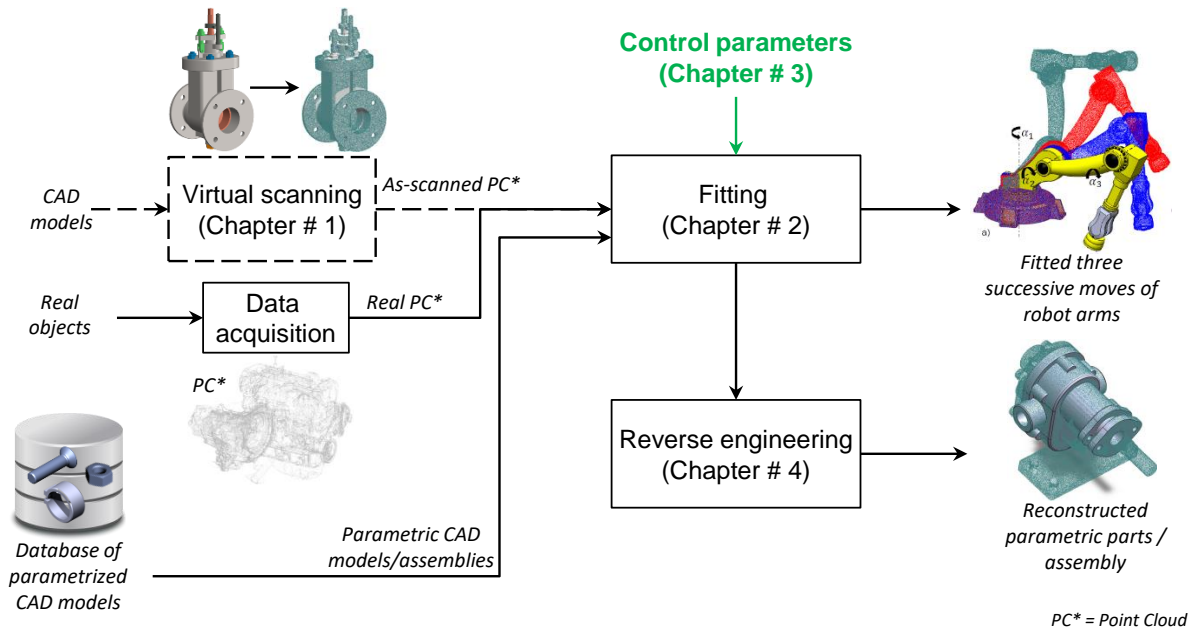


Figure 1 – Optimization of the Simulated Annealing-based fitting and the new reverse engineering technique for efficient reconstruction of the mechanical parts/assembly models.

Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications

Abstract

Due to its capacity to evolve in a large solution space, the Simulated Annealing (SA) algorithm has shown very promising results for the Reverse Engineering of editable CAD geometries including parametric 2D sketches, 3D CAD parts and assemblies. However, parameter setting is a key factor for its performance, but it is also awkward work. This paper addresses the way a SA-based Reverse Engineering technique can be enhanced by identifying its optimal default setting parameters for the fitting of CAD geometries to point clouds of digitized parts. The method integrates a sensitivity analysis to characterize the impact of the variations in the parameters of a CAD model on the evolution of the deviation between the CAD model itself and the point cloud to be fitted. The principles underpinning the adopted fitting algorithm are briefly recalled. A framework that uses design of experiments (DOEs) is introduced to identify and save in a database the best setting parameter values for given CAD models. This database is then exploited when considering the fitting of a new CAD model. Using similarity assessment, it is then possible to reuse the best setting parameter values of the most similar CAD model found in the database. The applied sensitivity analysis is described together with the comparison of the resulting sensitivity evolution curves with the changes in the CAD model parameters imposed by the SA algorithm. Possible improvements suggested by the analysis are implemented to enhance the efficiency of SA-based fitting. The overall approach is illustrated on the fitting of single mechanical parts but it can be directly extended to the fitting of parts' assemblies. It is particularly interesting in the context of the Industry 4.0 to update and maintain the coherence of the digital twins with respect to the evolution of the associated physical products and systems.

Keywords: Sensitivity analysis, CAD model parameters, Simulated Annealing, reverse engineering, digital twins

1. Introduction

Today, being able to reconstruct or update 3D representations of existing products and systems has become mainstream to support the creation and exploitation of digital twins in the context of Industry 4.0 [1]. The relative ease of access to more or less sophisticated 3D acquisition devices has certainly accelerated the demand while extending the range of possible applications. However, point cloud processing and Reverse Engineering (RE) of CAD models are not yet fully automatized. The resulting models are not fully exploitable by CAD software and the patch-by-patch reconstructions do not meet the Industry 4.0 requirements [2]. Most of the existing tools for RE ends up with non-editable dead models in which the structure and shape of the CAD models are frozen without any possibility to act on their parameter values that are even not accessible anyway. Only the position and orientation of the constitutive parts can be updated. Thus, current approaches fail to answer the update problem, and this is a major limitation which prevents the fast update of digital twins with respect to the evolution of the physical ones in the scope of the Industry 4.0. Moreover, not only they can be used to solve the update problem, but editable reverse engineered CAD geometries can also be modified in the later stages of the product development processes (PDP). In this paper, the update problem is transformed in a fitting problem for which the optimal parameter values of CAD models have to be found in order to minimize the deviation between the digital twins and the point clouds coming from the digitization of the

physical twins.

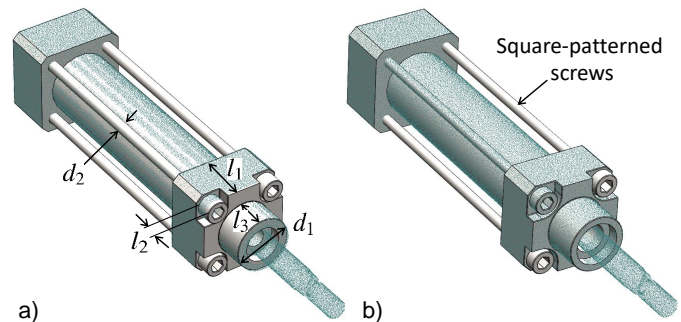


Figure 1: Fitting of a CAD assembly on the point cloud of a digitized pneumatic pump: coarse result without tuning the parameters of the SA-based fitting algorithm (a); and good-quality result after tuning (b).

There are few methods to reconstruct editable CAD geometries. Among them, metaheuristic algorithms like Simulated Annealing (SA) and Particle Swarm Optimization (PSO) have been cleverly used to optimize the parameters of the CAD geometries to fully fit in the point cloud obtained from data acquisition devices [3, 4]. These metaheuristic algorithms try to find global optimal solutions by both diversifying and intensifying the search in a large solution space. Even if such algorithms proved to be very beneficial to support the RE process, the tuning of their setting parameters is not an easy task. This paper addresses the way SA-based reconstructions can be enhanced

by identifying optimal default setting parameters for the fitting of CAD geometries to point clouds of digitized parts.

The RE process described in [3] considers as inputs the digitized point cloud, the CAD model to be fitted and the set of its parameters to be modified in order to minimize the deviation between the two. The applied metaheuristic SA algorithm has a stochastic behavior [5], with few knowledge on how the CAD parameter values are to be changed throughout the optimization. Therefore, if the SA setting parameters are not properly chosen, the fitting process may end up with either under or over-fitting of the CAD parts. Figure 1.a shows a bad fitting of a pneumatic pump consisting of multiple parts. Some control parameters of the CAD models that drive the shapes are under fitted (e.g. l_3) while other are overfitted (l_1, l_2, d_1 and d_2). To avoid bad quality fitting of CAD models and assemblies, a tuned SA-based fitting algorithm is used with setting parameters obtained following the tuning strategy proposed in this paper (Fig. 1.b). It is well-known that good parameter settings have a significant effect on the performance of an algorithm [6]. For example, SA algorithm is sensitive to the cooling factor, while a tabu search algorithm relies on a good choice of the tabu tenure [7]. A possible way to understand the influence between the output and parameters' evolution is sensitivity analysis [8, 9]. It has been used in various contexts, as, for example, for the parameter choices of SA for continuous network design [10], and various indices have been specified [11, 12, 13]. The sensitivity analysis allows the evolution of the influence of the CAD model parameters to be monitored throughout the fitting process. Comparing the obtained sensitivity evolution curves to the changes of the parameters directly imposed by the SA algorithm provides hints to improve the fitting process. Some basic and advanced parameters are required by the SA-based fitting algorithm to reverse engineer CAD models and, in this paper, a new framework is proposed that couples sensitivity analysis of the CAD parameters to SA algorithm and identifies the default parameters configurations with the help of a design of experiments (DoEs). Within this framework, a database is created to store together the CAD models with their best SA parameters configuration. Later, if a new CAD template is to be reverse engineered following this SA-based fitting technique, a case-based approach can be applied for the setting of the SA parameters by reusing the values stored for the most similar CAD model in the database.

The contribution of this paper is threefold: (i) a new framework using design of experiments (DoEs) for the creation of a database of CAD models stored together with their best SA-based fitting configurations; (ii) sensitivity analysis to study the influence of SA on the parameters of CAD models during the fitting process; (iii) identification of the default setting parameters for SA-based fitting to obtained best-fitted geometries using DoEs. The proposed strategy for tuning the SA parameters is generic in the sense that it could be also applied to other problems where SA is needed, and for which a database of best configurations could be set up for case re-use through similarity check.

The paper is organized as follows. Section 2 reviews some works related to sensitivity analysis as well as to the way

SA parameters can be tuned to improve its overall performance. SA-based fitting of CAD models is briefly discussed in section 3. Section 4 details the overall framework for the identification of best setting parameters to be used by the adopted SA-based fitting algorithm. The results are discussed in section 5 and the approach is validated on several examples including local and global fitting on real-scanned and virtually-generated point clouds. Lastly, section 6 concludes the paper and discusses future works.

2. Related works

Studies and experimentations have shown that the SA setting parameters, e.g. initial temperature, initial starting position and maximum number of iterations have a clear effect on the final fitting quality. Many attempts have been made to optimize the SA algorithm by tuning its setting parameters. For instance, Yang Jin et al. [14] studied the effects of parameters choice on the performance of SA algorithm to solve the continuous network design problem, but their findings and focus were mostly towards the temperature decrease rate and iteration number, without considering other parameters that also affect the quality of their SA-based optimization. Shi-hua Zhan et al. [15] proposed a new List-Based Simulated Annealing algorithm (LBSA) to solve the Traveling Salesman Problem (TSP) in which LBSA requires a novel list-based cooling schedule to control the decrease of temperature. In their method, the temperature list is adopted iteratively according to the topology of the solution space of the problem. The effectiveness and the sensitivity of the parameter of the list-based cooling schedule are illustrated on the TSP benchmark problem. In this method, a list of initial temperatures are generated and tested but again only temperature is considered to optimize the SA action towards the final solution.

Bellio et al. [16] used a single-stage SA optimization to solve the Curriculum-Based Course Timetabling (CB-CTT) problem. They proposed an extensive and statistically principled methodology for the parameters tuning procedure to model the relationships between the most relevant parameters of the solver and the features of the instance under consideration. The method tackles the parameters selection as a classification problem and builds a rule for choosing the set of parameters most likely to perform well for a given instance on the basis of given features. However, their study was limited to fewer instances to improve the previous results and out of 21 instances only 10 were improved. Some specific research has been devoted by Atiqullah et al. [17] to augment the convergence and related behavior of annealing algorithms by modifying its parameters, otherwise known as cooling schedule. They introduced an approach to tune the SA algorithm by combining algorithmic and parametric augmentations. They used a simpler cooling algorithm as compared to the complex schedules for designing a welded beam using their tuned up SA. Here again, a limited number of parameters is taken into account for the improvement of the SA algorithm, especially the cooling schedule.

In conclusion, it is obvious from the previous studies that SA-based optimization algorithms require some tuning of their

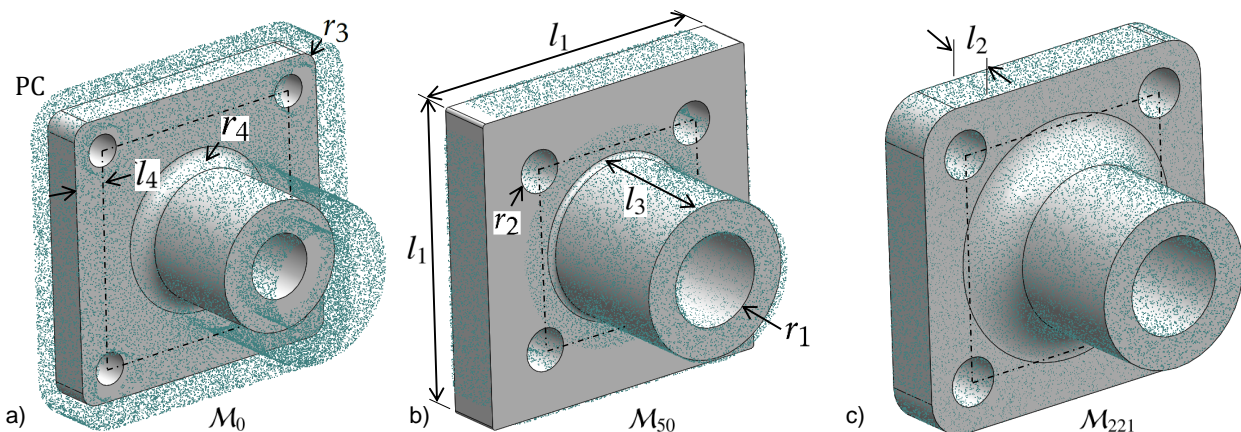


Figure 2: Global fitting of a rectangular flange defined by 8 control parameters: (a) initial configuration after the prearrangement step, (b) fitted part after 50 iterations of the SA algorithm, (c) final fitted part minimizing the deviation with the point cloud ($\varepsilon_s=10^{-1}$ and $M_{iter}=1000$).

setting parameters to get good-quality results in an efficient way. This is true for a wide range of applications, including the ones related to the RE of parametric CAD models. This paper presents a method to identify the best configuration of SA setting parameters when considering the optimization of CAD templates (e.g. parametric sketches, CAD parts and assemblies) to be fitted into point clouds of digitized parts. A framework is introduced to create a database of CAD parts stored together with their best SA parameters configuration. Then, when a new part is to be reconstructed using the SA-based fitting algorithm, the user can extract the best configuration settings from the available database using similarity checks between the CAD model to be fitted and the ones available in the database.

3. Simulated annealing-based fitting of CAD models

Recently, following a part-by-part rather than a patch-by-patch strategy, a breakthrough has been achieved for the reverse engineering of editable CAD models fitting point clouds of digitized mechanical parts or assemblies [3]. The idea is to work directly at the level of the part whose parameters are modified by an optimization algorithm until a good-quality fitting is obtained. To address this, a metaheuristic algorithm was required to handle a large solution space.

3.1. Simulated Annealing algorithm

Simulated Annealing has been widely used in different applications to solve optimization problems. The algorithm was designed for a 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 square of

distances between the tessellated CAD part \mathcal{M}^* and the point cloud PC) is minimized as shown in eq. (1). Other metaheuristics have been tested but have demonstrated a lower efficiency than SA. For instance, Particle Swarm Optimization (PSO) generates candidate solutions that can be significantly different from the initial position. This optimization strategy may result in configurations that are hard for the CAD modeler to update, and may even cause the software to crash.

To better understand the fitting technique, the method is exemplified on the fitting of a rectangular flange as shown in Figure 2. The input is the point cloud PC to which a parameterized CAD model \mathcal{M}_0 is to be fitted. The point cloud is composed of N points $PC[i]$, with $i \in [1..N]$, and the CAD model is parameterized by N_p control parameters p_k , with $k \in [1..N_p]$. Here, there are 8 control parameters overall: the lengths l_1 to l_4 , and the radii r_1 to r_4 . During the prearrangement step, the user locates in a coarse manner the part inside the point cloud, and chooses the control parameters of the CAD part to be optimized by the SA algorithm, this initializes the parameters values $p_{k,0}$ (Fig. 2.a). At each iteration j , the SA algorithm then tries to modify the parameters values $p_{k,j}$ until the updated CAD model perfectly fits the point cloud (Figures 2.b and 2.c). The quality of the fitting is assessed thanks to an energy function that characterizes the overall deviation between the point cloud and the CAD model. The process stops when the energy function no longer decreases with respect to a given accuracy ε_s , or when a max number of iterations M_{iter} is reached.

At an iteration $j \geq 0$ of the SA algorithm, the energy function to be minimized is evaluated as follows:

$$E(p_{1,j}, \dots, p_{N_p,j}) = \sum_{i=1}^N d^2(PC[i], \mathcal{M}_j^*(p_{1,j}, \dots, p_{N_p,j})) \quad (1)$$

wherein \mathcal{M}_j^* represents the tessellation of the CAD model \mathcal{M}_j after its update by the CAD modeler at the j -th iteration of the SA algorithm, and $d(\text{point}, \text{mesh})$ is the distance function that returns the closest distance between a *point* and a *mesh*. As detailed in [3], this process is further improved while allowing points of the PC to be filtered step after step, so as to allow

Categories	Parameters	Values
Mandatory	AnnealingFcn	annealingfast
	TemperatureFcn	temperatureexp
	StallLimit	25
	FunctionToler	1.00E-01
	InitialTemp	10
Advanced	InitialPosition	Position-02
	Weighting p_k	NoWeight
	Grouping p_k	NoGroup

Table 1: Values of the setting parameters split in two categories and gathered together in a so-called configuration C_κ used for fitting the rectangular flange.

local fitting of a part in the point cloud of a digitized mechanical assembly.

3.2. Setting parameters for SA algorithm

The adopted SA algorithm is controlled by certain setting parameters to allow the fitting of a CAD template in a given point cloud. The selection of the right setting parameters plays an important role to achieve a high-quality solution and it remains the crucial step that directly affects the final fitting results. Setting parameters are classified into two groups: the mandatory parameters and the advanced ones, which all together define a so-called parameters configuration C_κ . For instance, the parameter values used for the SA algorithm to reconstruct the rectangular flange of Figure 2 are shown in Table 1. Each parameter is detailed in the next sections. On one hand, the mandatory parameters are the essential parameters of SA and they must be specified to initiate the optimization. The final fitting results are highly influenced by their values, and they may produce unstable results if not selected wisely. On the other hand, additional parameters are introduced and result in a set of advanced tuning options, namely the weighting and grouping strategies. Each of them can be tuned differently depending on the fitting problem, and this can be tricky. Thus, in this paper, a methodology is developed to support the tuning of those parameters and the identification of a proper configuration C_κ . The idea is to build a database of best configurations known to produce quality results on several fitting examples, and then make use of a similarity assessment approach to find out the most similar fitting configuration with respect to a new CAD model to be fitted.

3.3. Description of the setting parameters

The setting parameters are briefly explained below, whereas the values they can take are presented when detailing the Design of Experiments that has been set up.

Regarding the mandatory parameters, annealing function (*AnnealingFcn*) is used by SA to generate a random trial point for the next point. The algorithm chooses the distance of the trial point from the current point by a probability distribution with a scale depending on the current temperature. Temperature function (*TemperatureFcn*) is used to update the temperature. The algorithm systematically lowers the temperature, storing the best point found so far. Maximum stall iterations (*StallLimit*) is used as one of the stopping criteria. SA keeps track

of the average change in the function value for *StallLimit* iterations and the optimization process stops if the SA does not find any better solution point within the *StallLimit*. Function tolerance (*FuncToler*) is another stopping criterion of SA according to which algorithm runs until the average change in the value of the objective function is less than the value of *FuncToler*. Initial temperature (*InitialTemp*) controls the overall search results for the next candidates of parameters p_k . For each dimension, the temperature is used to limit the extent of search in that dimension. Finally, the *InitialPosition* of the CAD model also has influence on the final result as SA chooses the distance of the trial points from the initial values of the parameters p_k by a probability with a scale depending on the initial temperature.

As for the advanced parameters, a weighting strategy for the parameters p_k can be used to avoid under and overfitting of small features like fillets and chamfers which in some cases are neglected by the SA algorithm due to a small contribution in the overall energy (see section 4.5). In addition, a grouping strategy of parameters p_k can be selected as another mean of avoiding under and over-fitting of small features especially when there are many parameters controlling the CAD geometry. Grouping of parameters p_k can be done according to their sensitivities (section 4.3) or according to the type of features they refer to (section 4.4). For the cases where the number of parameters is small, one single group can be considered.

4. Framework for the selection of default configuration for SA-based fitting

This section introduces the framework developed to support the identification of the best configuration C_κ for a given CAD model to be fitted in a point cloud. The idea is to first build a database of fitting examples storing both the CAD models together with their associated best parameter configurations. Then, for a new CAD model to be fitted, it is possible to refer to the database by means of a similarity assessment [18], to find out and then use the best configuration associated to the most similar fitting example.

Thus, the framework can be decomposed in two parts: the one used for the generation of the database (Fig. 3), and the one used to look for a similar configuration when a new CAD model is to be fitted (Fig. 9). This two-part framework is generic and could be used in other contexts. To create such a database, CAD models are fitted using SA annealing algorithm while playing with each setting parameter to get a good quality fitting and identify the associated best parameters configuration. Various parts are globally and locally fitted to their respective point clouds, obtained either from a laser scanner or virtually generated to get as-scanned point clouds [19]). Many possible configurations of the setting parameters for the SA algorithm are tested using DoEs to identify and store the best configuration in the database of CAD models. For simplicity, the first part of the framework has been explained on the fitting of a rectangular flange (Fig. 2). Thus, subsections 4.1 to 4.7 explain how to identify best configurations and populate the database, whereas subsection 4.8 details how to use the database, in a second step

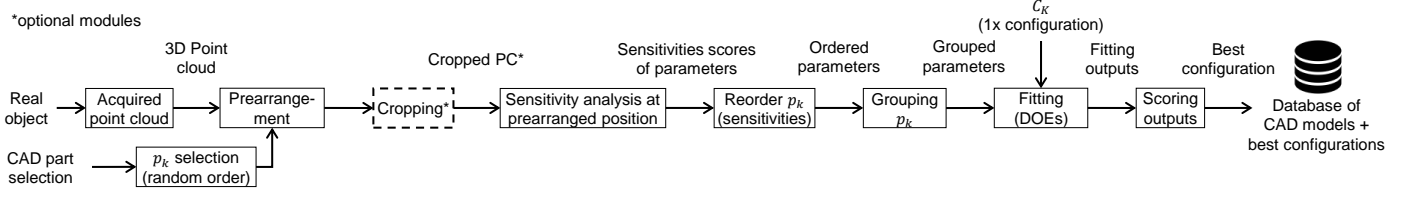


Figure 3: Overall framework used to populate a database of CAD models stored together with their best configurations C_k for the SA.

and for an unknown CAD model to be fitted, to retrieve the best configuration associated to the most similar fitting example.

4.1. Prearrangement and cropping

Following the framework of Figure 3, the process of adding an instance to the database starts with the selection of the initial parameterized CAD model M_0 to fit in a given point cloud PC obtained after data acquisition. In random order, all the parameters p_k controlling the shape of the CAD model are selected for the optimization. Next, the CAD model is prearranged with the given point cloud PC (Fig. 2.a) and the point cloud can optionally be segmented if the intended optimization case is for the local fitting. This happens when treating assemblies, where the template corresponds to a sub-part of the acquired object and thus only a segment of the point cloud take part in the optimization process, otherwise, the whole point cloud is utilized for the global fitting as required when the template geometry corresponds to the whole object acquired.

4.2. Evolution of parameters p_k during SA-based fitting

In order to understand how SA algorithm modifies the values of the parameters step after step, the evolution of the parameters is tracked throughout the fitting process. This can be done either in an absolute or relative manner, as follows:

$$\delta p_{k,j} = p_{k,j+1} - p_{k,j} \quad (2)$$

$$\Delta p_{k,j} = \frac{\delta p_{k,j}}{p_{k,j}} \quad (3)$$

Thus, the value $\delta p_{k,j}$ characterizes the absolute evolution of the parameter p_k between iterations j and $j+1$, whereas $\Delta p_{k,j}$ refers to its relative evolution. At each iteration j of the SA algorithm, it is therefore possible to study the evolution of each parameter p_k , and see the decision taken by the SA algorithm on how to modify the parameter value for the next step $j+1$. This is illustrated in Figure 4 for the fitting of the rectangular flange and using setting parameters for the SA listed in Table 1. From these evolution curves, it can be seen that all the parameters are modified more or less in the same manner.

Following the configuration C_k of Table 1, the final absolute δp_k^F and relative deviations Δp_k^F of the parameters are computed and the results are listed in Table 2:

$$\Delta p_k^F = \frac{|\delta p_k^F|}{|p_k^D|} = \frac{|p_k^D - p_k^F|}{|p_k^D|} \quad (4)$$

where p_k^F are the final values of the parameters controlling the fitted CAD model, and p_k^D are the original values of the ground

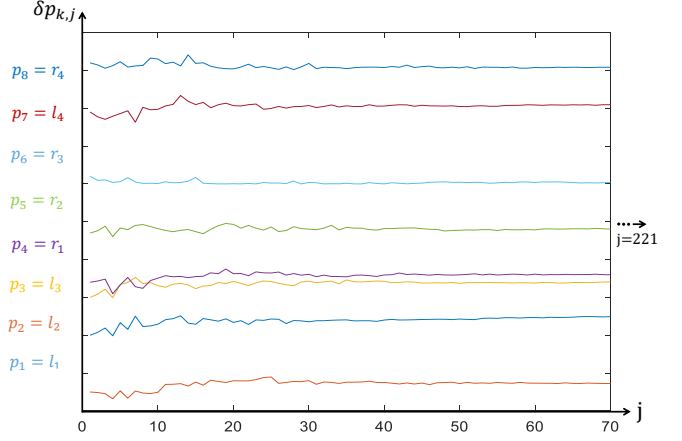


Figure 4: Changes of $\delta p_{k,j}$ imposed by the SA algorithm at each iteration j of the fitting process on the rectangular flange.

truth CAD model that has been used to virtually generate the given point cloud [3]. The deviations of some parameters are higher, e.g. the two radii r_3 and r_4 . This is due to the fact that SA neither takes into account the type of parameters nor how much they contribute to the overall energy of eq. (1). To tackle this problem a sensitivity analysis of parameters is required, which is detailed in the next section.

Groups	p_k	p_k^0 (mm)	p_k^D (mm)	p_k^F (mm)	δp_k^F (mm)	Δp_k^F
G_1	l_2	5	7	6.8313	0.17	0.0241
	l_3	15	20	19.975	0.02	0.0012
	r_1	4	6	6.0901	-0.09	0.0150
	l_1	35	40	39.719	0.28	0.0070
	r_3	2	5	3.4563	1.54	0.3087
	r_2	2	3	3.1657	-0.17	0.0552
	l_4	2	3	3.0214	-0.02	0.0071
	r_4	2	5	3.6982	1.30	0.2604

Table 2: Results for the global fitting of rectangular flange following the configuration C_k of Table 1.

4.3. Sensitivity analysis

The SA algorithm used for fitting of CAD models works as a black box and neither takes into account the semantics associated to the various parameters p_k nor the way their variations may influence the evolution of the energy function step after step. Indeed, while acting on the variables to be optimized, the SA algorithm does not distinguish for instance parameters like

length, width, thickness and radius that characterize the features, and all the variables are considered in the same way. It also does not take into account the fact that the influence of a parameter variation on the energy may vary a lot depending on whether the parameter is associated to a structural feature (e.g. the length l_1 of the rectangular flange in Figure 2.b) or to a detail feature (e.g. the radius r_4).

To better compare the influence of the parameters, a sensitivity estimator $S_{k,j}$ is here introduced to compute at each step j of the optimization process the sensitivity of the energy function with respect to each parameter p_k :

$$S_{k,j} = \frac{\partial E}{\partial p_k} |_{(p_{1,j}, \dots, p_{N_{p,j}})} \text{ with } j \geq 0 \quad (5)$$

Since the energy function is computed using the nearest point algorithm, it cannot be differentiated. Thus, the sensitivities are approximated using a first-order finite difference scheme (either forward or backward) directly controlled by the step h_k chosen for the corresponding parameter p_k :

$$S_{k,j}^+ = \frac{E(p_{1,j}, \dots, p_{k,j} + h_k, \dots, p_{N_{p,j}}) - E(p_{1,j}, \dots, p_{k,j}, \dots, p_{N_{p,j}})}{h_k} + O(h_k), \text{ with } j \geq 0 \quad (6)$$

$$S_{k,j}^- = \frac{E(p_{1,j}, \dots, p_{k,j}, \dots, p_{N_{p,j}}) - E(p_{1,j}, \dots, p_{k,j} - h_k, \dots, p_{N_{p,j}})}{h_k} + O(h_k), \text{ with } j \geq 0 \quad (7)$$

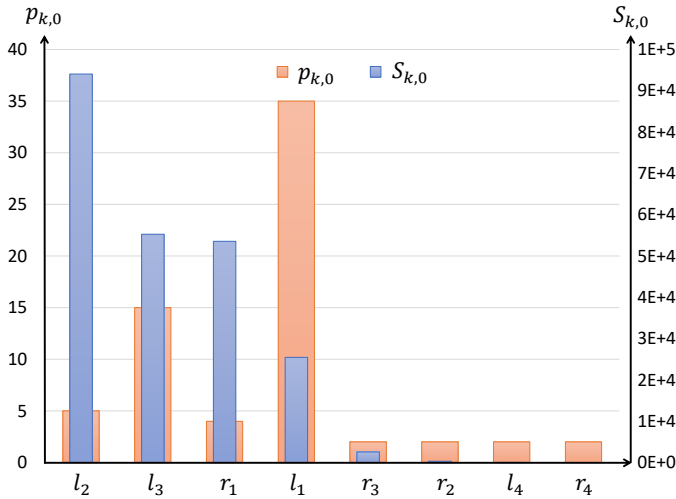


Figure 5: Sensitivities $S_{k,0}$ of the 8 parameters p_k of the rectangular flange at prearranged position ($j = 0$).

Here, the term $O(h_k)$ represents the order of magnitude of the approximation. More specifically, performing such a sensitivity analysis using all the parameters of the CAD model at the prearranged position ($j = 0$) returns the sensitivities $S_{k,0}$ of each

parameter. This shows how much each parameter p_k affects the change of the overall energy. For example, the sensitivity analysis of the rectangular flange having 8 control parameters is performed at its prearranged position (Fig. 2.a) and the resulting forward sensitivity against each parameter p_k is shown in Figure 5. Thus, parameters l_2 , l_3 , r_1 and l_1 are the ones that modify the global energy E the most, while the influence of r_2 , l_4 and r_4 is almost negligible. It can also be seen that sensitivities are independent of the magnitude of the parameter values $p_{k,0}$ at prearranged position. After the computation of the sensitivities, parameters are rearranged in a decreasing order according to their sensitivity level (Fig. 5), and this helps grouping the parameters as discussed in section 4.4.

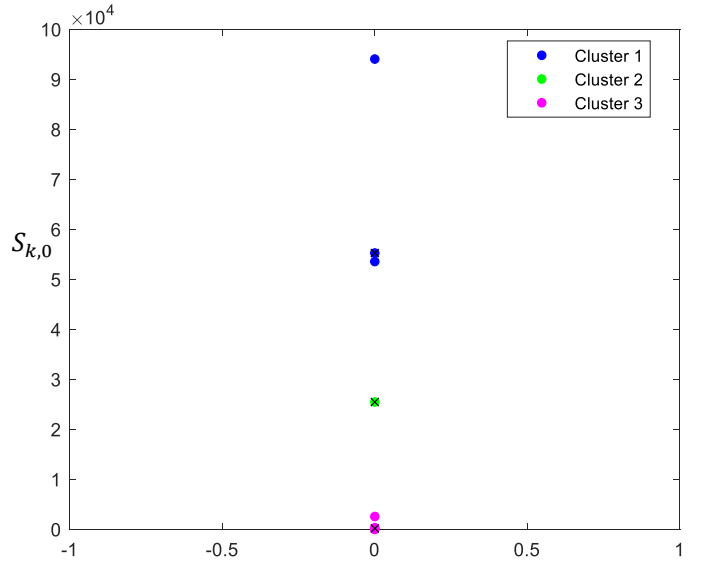


Figure 6: Grouping parameters at prearranged position ($j = 0$) according to sensitivity levels and using K-means clustering technique ($K = 3$).

4.4. Grouping strategies for parameters p_k

As part of the advanced SA settings, parameters p_k can optionally be clustered in several groups. Actually, grouping the parameters allows for a better and more stable fitting, and it also reduces the risk of getting the optimization process stuck in a local minimum. Conversely, considering all parameters in a single group may result in under or overfitting of small features. Thus, three grouping strategies are proposed:

- *Grouping according to sensitivity level* can be performed in an automatic manner using the K-means clustering technique. In this case, the sensitivity values are directly used as input of the K-means clustering algorithm that partitions the points into a number K of clusters. The resulting partitions minimize the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. By default, K-means uses squared Euclidean distances. For the rectangular flange, this grouping strategy then returns three groups as shown in Figure 6. The max value for the number K of clusters is assumed as:

$$K = \min\{3; N_p\} \quad (8)$$

- *Grouping according to the level of details* of the shape can be performed in a semi-automatic manner using a three-level organization, and following the commonly adopted CAD modeling strategies: G_1 groups the parameters of the assembly structure as well as the ones driving the structural features (e.g. pockets, revolutions), G_2 gathers the parameters of the detail features (e.g. holes, ribs) and G_3 includes the parameters used to finalize the shapes (e.g. fillets, chamfers).
- *NoGroup* strategy can be adopted by considering all the parameters in a single group.

4.5. Weighting strategies

As part of the advanced SA settings, different weighting strategies can be followed to allow heterogeneous scaling on the parameter values as the SA algorithm goes on. Indeed, the trial points generated by *annealingFcn* basically apply the same scale of distribution of distances on all parameters p_k , independently of the type of feature they refer to. This is considered as a *NoWeight* weighting strategy. As a consequence, in some cases, due to small contributions in the overall energy, the energies related to small features like fillets and chamfers are neglected by the SA algorithm and this may cause under and overfitting of such features. To avoid this, weights can be put on the parameters at each iteration j of the fitting process. Thus, after performing the sensitivity analysis at the prearranged position (section 4.3), the parameters are rearranged according to the decreasing order of their sensitivity levels (first parameter being the most sensitive and the last as the least one). Then, the idea is to cleverly change the scale distribution of *annealingFcn* that is generally uniform with equal weightage for all the parameters. To do so, two strategies are here proposed. To give more weightage to the least sensitive parameters, *LinearWeights* can be applied to all the parameters. The first parameter being the highest sensitive is given a weightage value of 1 (no weight) and for the trailing parameters 5% increment is added (i.e. 1.05 weight for the second parameter and 1.10 for the third and so on). In this way, the last parameter (least sensitive) has the highest weight, which helps the evolution of the parameter not to remain blocked at the local minimum. Alternatively, individual weights or *SelectedWeights* can be provided to the least sensitive parameters. In the proposed method, *SelectedWeights* are only given for fillets and chamfers, which weight 50% more than the other parameters to avoid local minimum.

4.6. Design of experiments set up

A configuration of settings C_k is composed of 8 setting parameters (6 mandatory and 2 advanced), each of which influencing the final quality of fitting. Thus, for a given CAD model to be fitted, testing all possible combinations of those parameters would result in a significantly large number of experiments to be conducted. To overcome this issue, a limited number of experiments is conducted by setting up a Design of Experiments (DoEs) that focuses on a restricted set of configurations C_k , among which the best configuration is to be identified for each sample of the database. Taguchi orthogonal array technique has been used to conduct a minimum number of

experiments [20]. All the setting parameters with three possible levels are listed in Table 3. One of the parameter, namely *AnnealingFcn*, has 2 levels (2^1) whereas the others have 3 levels (3^7). According to the number of parameters and the number of levels, the $L_{18}(2^1, 3^7)$ table of Taguchi orthogonal array has been chosen [21]. During each batch of 18 experiments, the quality of the fitting is observed on the basis of three criteria introduced in the next subsection.

More precisely, to help SA in setting the trial points distance distribution, *AnnealingFcn* can rely on two options, either *annealingfast* or *annealingboltz* [22, 23]. *Annealingfast* function generates a point equal to the current point plus a random step of size equal to the current temperature. While *annealingboltz* function generates a point based on the current point and the current temperature using a multivariate normal distribution. The *TemperatureFcn* can be customized with three options: *temperatureexp*, *temperaturefast* or *temperatureboltz* [22, 23]. In the *temperatureexp* scheme, the temperature at any given step is 0.95 times the temperature at the previous step. This causes the temperature to go down slowly at first but ultimately get cooler faster than other schemes. If another scheme is desired, e.g. *boltzman* schedule or *fast* schedule annealing, then *temperatureboltz* or *temperaturefast* can be used respectively. For the termination criteria, *StallLimit* is provided with three values of 10, 25, and 50 iterations. For all three values, the SA algorithm will be tested if it can find the next better solution of parameters p_k from the current position, within given limits without spending too many iterations. For instance, 10 iterations will provide a small window for the SA to find the next best solution, while 50 iterations will provide sufficient time but it also increases the overall optimization time. *FunctionToler* is a second stopping criterion that can also be customized with three input values as shown in Table 3. The values are closed to zero to force the algorithm to find new best values of parameters p_k until the overall energy returned by the objective function is equal to one of these three values.

Parameters	level ₁	level ₂	level ₃
AnnealingFcn	annealingboltz	annealingfast	-
TemperatureFcn	temperatureexp	temperatureboltz	temperaturefast
StallLimit	10	25	50
FunctionToler	0e+00	1e-01	1e-06
InitialTemp	5	10	15
InitialPosition	Position-01	Position-02	Position-03
Weighting p_k	LinearWeights	SelectedWeights	NoWeight
Grouping p_k	Group (Sensitivity)	Group (Features)	NoGroup

Table 3: List of setting parameters with corresponding three levels ($level_{1-3}$) used for setting up the DoEs.

The selection of the initial temperature T_0 follows the method suggested by [3], i.e. while looking for the optimal initial temperature within a given window of temperatures, and the one which provides the least energy is chosen. For all the CAD models added in the database, 10 is identified as the best initial temperature, and to conduct DoEs, two more values are chosen around the best-identified temperature by putting a 5 degrees change on the positive and negative sides to test different tem-

C_k	AnnealingFcn	TemperatureFcn	StallLimit	FunctionTolerance	Grouping p_k	Weighting p_k	InitialTemp	InitialPosition	Scores
1	annealingboltz	temperatureexp	10	0.00E+00	NoGroup	NoWeight	5	Position-01	25
2	annealingboltz	temperatureexp	25	1.00E-01	Group(Features)	LinearWeights	10	Position-02	21
3	annealingboltz	temperatureexp	50	1.00E-06	Group(Sensitivity)	SelectedWeights	15	Position-03	19
4	annealingboltz	temperatureboltz	10	0.00E+00	Group(Features)	LinearWeights	15	Position-03	16
5	annealingboltz	temperatureboltz	25	1.00E-01	Group(Sensitivity)	SelectedWeights	5	Position-01	25
6	annealingboltz	temperatureboltz	50	1.00E-06	NoGroup	NoWeight	10	Position-02	12
7	annealingboltz	temperaturefast	10	1.00E-01	NoGroup	SelectedWeights	10	Position-03	15
8	annealingboltz	temperaturefast	25	1.00E-06	Group(Features)	NoWeight	15	Position-01	16
9	annealingboltz	temperaturefast	50	0.00E+00	Group(Sensitivity)	LinearWeights	5	Position-02	18
10	annealingfast	temperatureexp	10	1.00E-06	Group(Sensitivity)	LinearWeights	10	Position-01	25
11	annealingfast	temperatureexp	25	0.00E+00	NoGroup	SelectedWeights	15	Position-02	22
12	annealingfast	temperatureexp	50	1.00E-01	Group(Features)	NoWeight	5	Position-03	16
13	annealingfast	temperatureboltz	10	1.00E-01	Group(Sensitivity)	NoWeight	15	Position-02	23
14	annealingfast	temperatureboltz	25	1.00E-06	NoGroup	LinearWeights	5	Position-03	18
15	annealingfast	temperatureboltz	50	0.00E+00	Group(Features)	SelectedWeights	10	Position-01	24
16	annealingfast	temperaturefast	10	1.00E-06	Group(Features)	SelectedWeights	5	Position-02	18
17	annealingfast	temperaturefast	25	0.00E+00	Group(Sensitivity)	NoWeight	10	Position-03	13
18	annealingfast	temperaturefast	50	1.00E-01	NoGroup	LinearWeights	15	Position-01	15

Table 4: Fitting of rectangular flange for 18 configurations C_k ($k \in [1..18]$) of DoEs using L_{18} table of Taguchi orthogonal array.

peratures. Hence, three initial temperatures are chosen for the DoEs, i.e. 5, 10, and 15. Analogously, for the initial position, three initial values are selected for the DoEs. All three initial positions are chosen by roughly positioning the CAD model close to the point cloud and by modifying the control parameters p_k of the rectangular flange to have a percentage volume difference of roughly 20%, 15% and 10% for Position-01, Position-02 and Position-03 respectively. This also initializes the values of the control parameters of the CAD models at initial position p_k^0 , values that are then used for the initialization of the optimization process. Because of the random nature of the SA, the results obtained from DoEs can be biased. Indeed, a SA run with a certain set of parameters can produce exceptional results compared to what it would produce most of the time. In order to smooth this effect, for each DoE, SA is run three times and the mean of the three output measurements is finally computed.

SA-based optimization. Each of the 18 configurations C_k is evaluated on the basis of three criteria: (i) the number of iterations j_{max} taken by SA to find the optimal solution, (ii) the final energy $E_{|j=j_{max}}$ obtained at the end of the optimization, (iii) the sum of the final absolute deviations $\Delta p_{k,j_{max}}$ of all parameters. Figure 7 shows the results obtained when running the 18 configurations on the rectangular flange example. For sake of clarity, the results for the three criteria are normalized and represented as a percentage using C_1 as a reference. Thus, for C_1 , each criterion is evaluated as 100% and the rest of the configurations are compared to it for normalization. It can be clearly seen that each configuration C_k ($k \in [1..18]$) has different outcomes. For example, the result of C_7 shows a smaller sum of absolute deviations, with less number of iterations but with a higher final energy. While C_{17} shows opposite results where it took more iterations to complete the fitting, with better results in terms of absolute deviations and final energy.



Figure 7: Results of the DoEs for the rectangular flange normalized taking C_1 as a reference.

4.7. Scoring DoEs results for the best configuration selection

To identify the best possible configuration for the SA-based fitting algorithm, a scoring approach is used to rank the 18 configurations involved in the DoEs. Table 4 shows all the configurations and their corresponding input values used for the

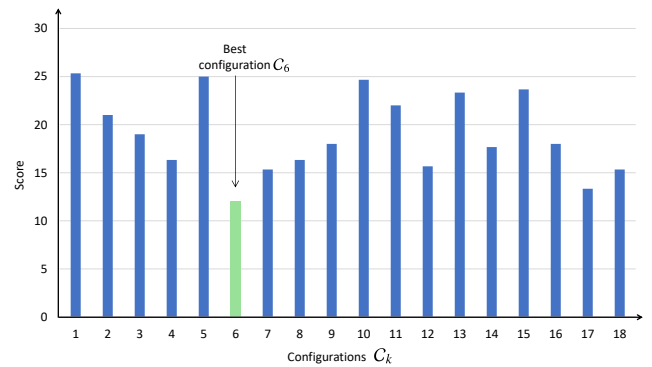


Figure 8: Scores of DoEs result for the rectangular flange.

Then, to identify the best configuration among the 18 for a given fitting example, the three quality criteria, i.e. *FinalEnergy*, *Sum Abs.Dev* and *Iterations* are ranked in an ascending order starting from the lowest to the highest value. To compute the final score of a configuration, the independent ranks of the three quality criteria are then sum up, and the configuration de-

picking the minimum score is considered as best configuration for this fitting example. Figure 8 shows the scores related to each configurations of the DoEs run on the rectangular flange.

For the rectangular flange, configuration C_6 appears to be the best configuration as it shows the lowest score of 12 with its setting parameters highlighted in green in Table 4. In this particular case, *annealingboltz* with *temperatureboltz* for SA annealing produced the best result of the fitting. The *StallLimit* is 50 which means the algorithm will have about 50 iterations to look for new best values of parameters, otherwise the algorithm will terminate. *StallLimit* with 50 iterations will give sufficient time to the algorithm to search for the next best set of parameters p_k . Furthermore, as this example is rather simple, *NoWeight* and *NoGroup* are needed, and Position-02 is sufficient to start the optimization process.

Finally, once this framework defined, a database can be created and populated while storing CAD models together with their best configurations. Hence, as discussed in the next subsection, the values stored in the database can be used as references to configure the SA setting parameters when considering a new CAD model to be fitted in a point cloud using the adopted SA-based fitting technique.

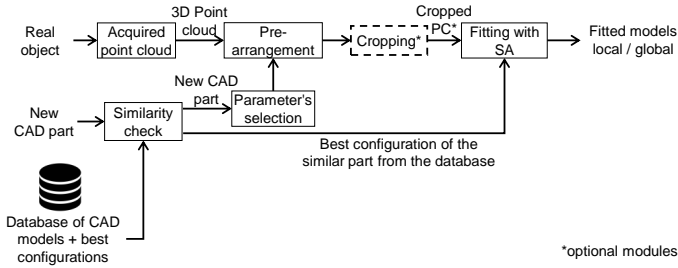


Figure 9: Overall framework for fitting a new CAD model using the best configuration of a similar CAD model found in the database.

4.8. Fitting new CAD template reusing an acquired knowledge

As introduced at the beginning of section 4, once created a database gathering CAD models together with their best configurations for point cloud fitting, if the user wants to reconstruct a new CAD model, the second framework illustrated in Figure 9 should be followed. The idea is to make use of a similarity assessment tool to identify the CAD model of the database which resembles the most to the new fitting case, and then use its best configuration to tune the SA setting parameters. In this work, the comparison tool of Giannini, et al. [18] has been used to assess the similarity but other tools for the retrieval of similar CAD models could also be used. As terms of results, this tool returns a similarity matrix that helps in the identification of the CAD model that has the closest geometric description with the new CAD geometry to be fitted. Once the most similar geometry identified from the database, the associated best configuration can be used for the new fitting. The user still needs to specify which control parameters p_k of the new CAD model should be considered as variables of the optimization process. At the end, using this best configuration extracted from the database,

the SA-based fitting algorithm will tend to produce results depicting low deviations for the parameters p_k , low overall final energy as well as a limited number of iterations.

5. Results and discussion

The core of the fitting algorithm has been implemented in MATLAB®, which is able to call the built-in functions of SolidWorks® to perform the successive CAD model updates and ensure the consistency of the resulting B-Rep model during the optimization loops.

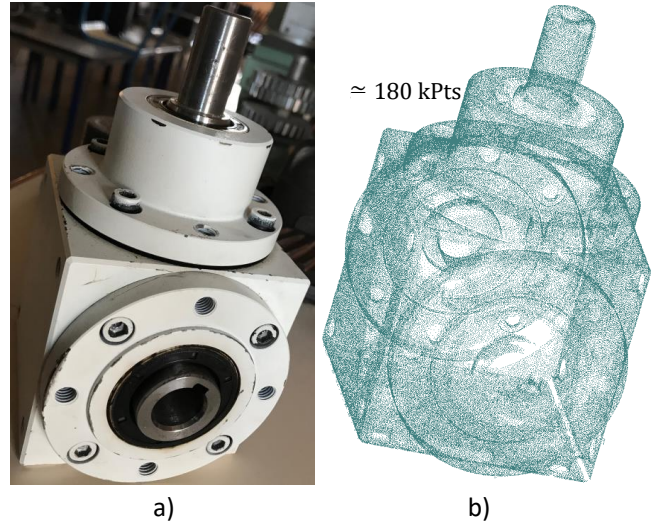


Figure 10: Point cloud (b) obtained while scanning a real gearbox (a) without disassembling it and using a ROMER Absolute Arm 7520 SI laser scanner.

5.1. Creation of a database of CAD models

To create the database and start populating it with CAD models, 17 CAD models have been selected (Fig. 14). Due to space limitation, in this paper, only four test cases from the database are detailed, but the similarity assessment is performed on all of them. A rectangular flange, a gland, a chain link and a round flange, all of them being fitted several times to their respective point clouds using our SA-based fitting algorithm through DoEs (section 4.6). The considered CAD templates before and after the fitting with their best configurations are shown in Figure 11. The point clouds for the rectangular flange, chain assembly and the gland are virtually generated using as-scanned point cloud generation technique [19]. Global fitting is to be considered for the rectangular flange, and local fitting for the gland and chain link. The fourth example considers the local fitting of a round flange into a real point cloud obtained from scanning a gearbox assembly (Fig. 10.a). Here, 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 it prior to its digitalization. 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 having about 180k points is obtained (Fig. 10.b).

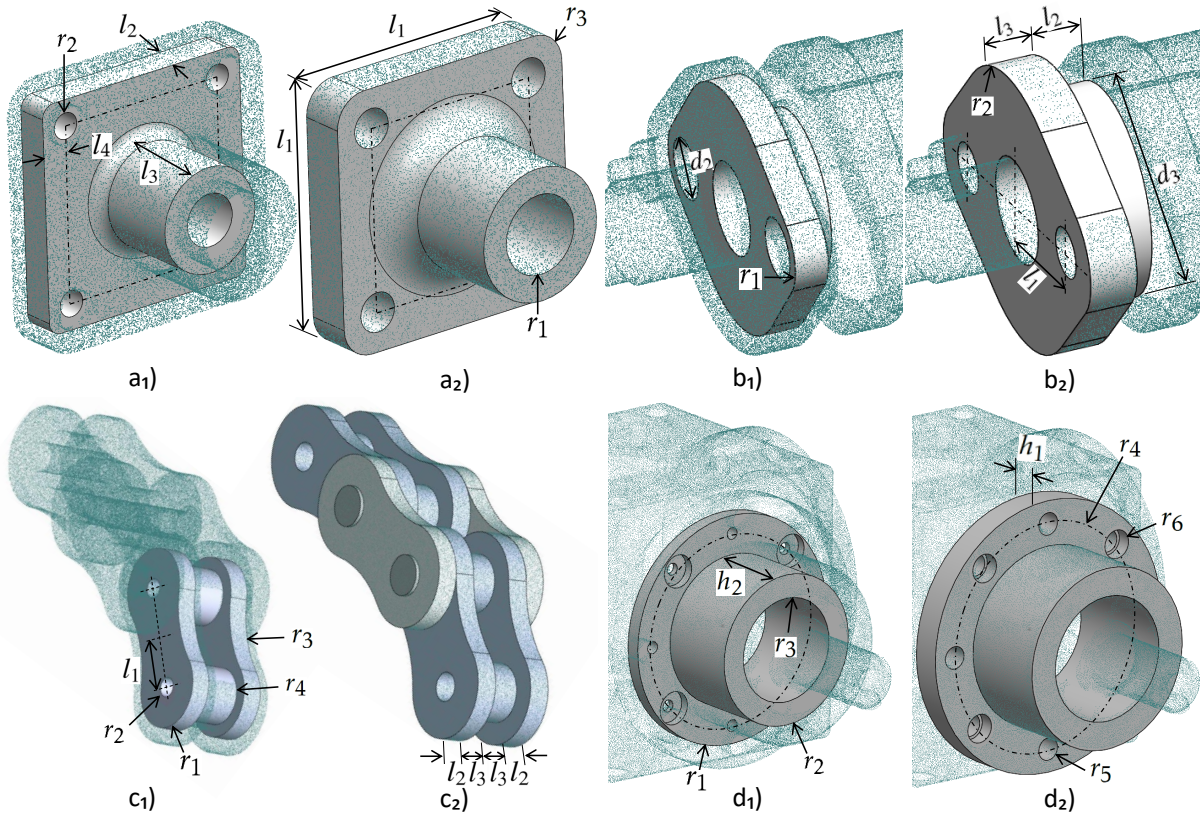


Figure 11: Four CAD models before and after the fitting with their best configuration resulting from the DoEs: (a₁) coarse pre-arrangement of the rectangular flange in the virtually-generated point cloud; (a₂) globally fitted rectangular flange; (b₁) pre-arrangement of the gland; (b₂) locally fitted gland in the virtually-generated point cloud of a chain assembly; (c₁) pre-arrangement of the chain link; (c₂) locally fitted chain link in the virtually-generated point cloud of a chain assembly; (d₁) pre-arrangement of the round flange; (d₂) locally fitted round flange in the real scanned point cloud of a gearbox assembly.

Once the CAD models are pre-arranged (Figures 11.a₁, b₁ and c₁), their sensitivities are computed at prearranged position according to steps detailed in section 4.3. All the steps h_k used to compute the sensitivities are equal and only depend on the type of parameter they are associated to. Thus, step $h_k=0.3\text{mm}$ in case p_k is a length, and $h_k=0.3^\circ$ in case it is an angle. Those values have been chosen in coherence with the accuracy of the adopted CAD modeler. For the sensitivities at pre-arranged position $S_{k,0}$, the forward finite difference scheme $S_{k,j}^+$ has been used. After sensitivity analysis at the prearranged position, the control parameters p_k are rearranged in a decreasing order starting from the most sensitive to the least sensitive one (Fig. 5), and this is repeated for all the CAD models at the pre-arranged positions. Then, the parameters are grouped according to the different criteria detailed in section 4.4. For all the CAD models used for the creation of the database, Table 5 shows the grouping of parameters according to either the level of sensitivities or the type of features they belong to.

The next step is to fit the CAD models with the SA-based fitting algorithm using DoEs to test the 18 configurations of setting parameters and identify the best configuration C_k for each CAD model. The rectangular and round flanges have 8 control parameters that drive their shape during the optimization process, while the gland and the chain link are controlled by 7 pa-

Groups	According to sensitivities using K-means	According to features
Rectangular flange		
G_1	l_2, l_3, r_1	l_2, l_3, r_1, l_1
G_2	l_1	l_4, r_2
G_3	r_3, r_2, l_4, r_4	r_3, r_4
Gland		
G_1	l_3, l_2, r_2, r_1, l_1	l_3, l_2, r_2, r_1, d_3
G_2	d_3, d_2	l_1, d_2
Chain link		
G_1	r_1, r_4, l_2	r_1, r_4, l_2, l_1, l_3
G_2	l_1, l_3, r_2	r_2, r_3
G_3	r_3	–
Round flange		
G_1	h_1, r_1, h_2, r_2	h_1, r_1, h_2, r_2
G_2	r_4, r_5	r_4, r_5, r_6, d_1
G_3	r_6, d_1	–

Table 5: Grouping of parameters p_k according to their sensitivity levels and to the types of feature they refer to.

rameters each, as visible in Figure 11 that also shows the fitting results using best configurations. All the configurations C_k of parameters used for the fitting are listed in Table 4. After completing all the experiments, the best configuration of each CAD

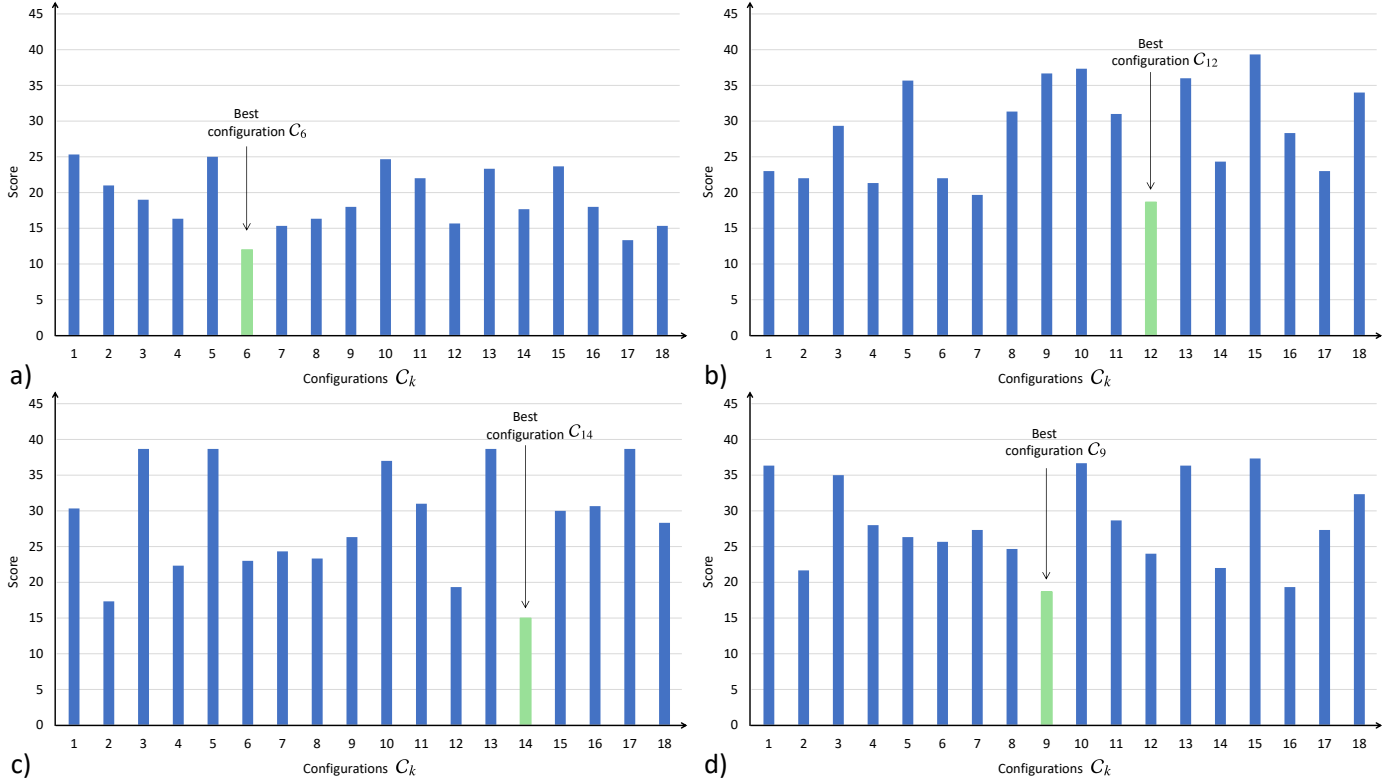


Figure 12: Scores obtained from averaging the results obtained from DoEs performed 3 times for the selection of best configuration of: (a) rectangular flange; (b) gland; (c) chain link and (d) round flange.

model is identified by selecting the configuration with the lowest sum of scores between the resulting final energy, the sum of absolute deviations of parameters p_k and the total number of iterations. Obtained results are depicted in Figure 12 and clearly show the stochastic behavior of SA that is not very consistent for the fitting of all four models and cannot be correlated. In order to smooth this effect, each set of DoEs has been run three times and mean scores are taken. All the identified best configurations 6, 12, 14, and 9 are stored in the database together with the related CAD models. The corresponding setting parameters are listed in Table 4.

5.2. Fitting new CAD templates using stored configurations of similar parts

This section details how a new CAD template can be fitted using the stored best configuration of the most similar part retrieved by performing a similarity analysis. This is illustrated on the fitting of a bend joint (Fig. 13). The global similarity assessment has been performed between the bend joint (considered as query model) and the 17 models in the database using the tool of Giannini et al. [18]. They adopted an attributed graph containing both geometrical and topological information to represent a model. A correspondence among two models is found by the identification of the maximum common subgraph between the two attributed graphs representing the models. This problem is solved transposing to the maximum clique (MC) problem employing an association graph in which the arcs linking the associated faces vary according to the search objectives. To

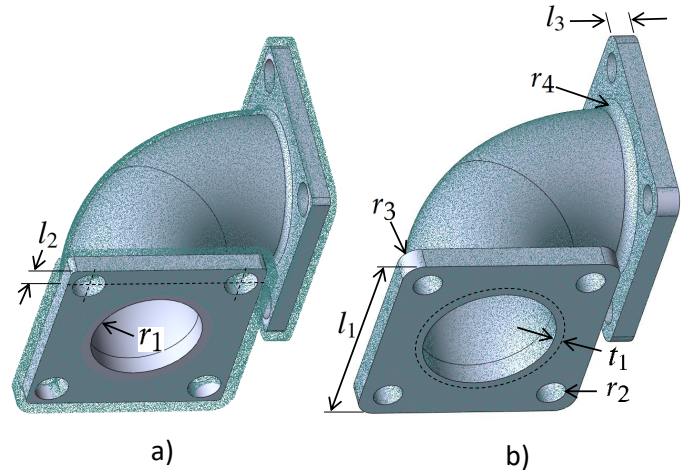


Figure 13: Global fitting of bend joint defined by 8 control parameters: (a) initial configuration after the prearrangement step, (b) final fitted part minimizing the deviation with the point cloud and using best configuration C_6 .

detect the MC, a simulated annealing-based minimization is used to minimize the function characterizing the clique property. Figure 14 shows all the CAD models in the database used for similarity check and the results of this analysis on the new bend joint is presented in Table 6. The number of corresponding nodes and similarity measure for each model are also listed. The query model is composed of 34 faces, of which 10 are planar, 10 cylindrical, 10 fillets, and the 4 are sweep parallel faces of

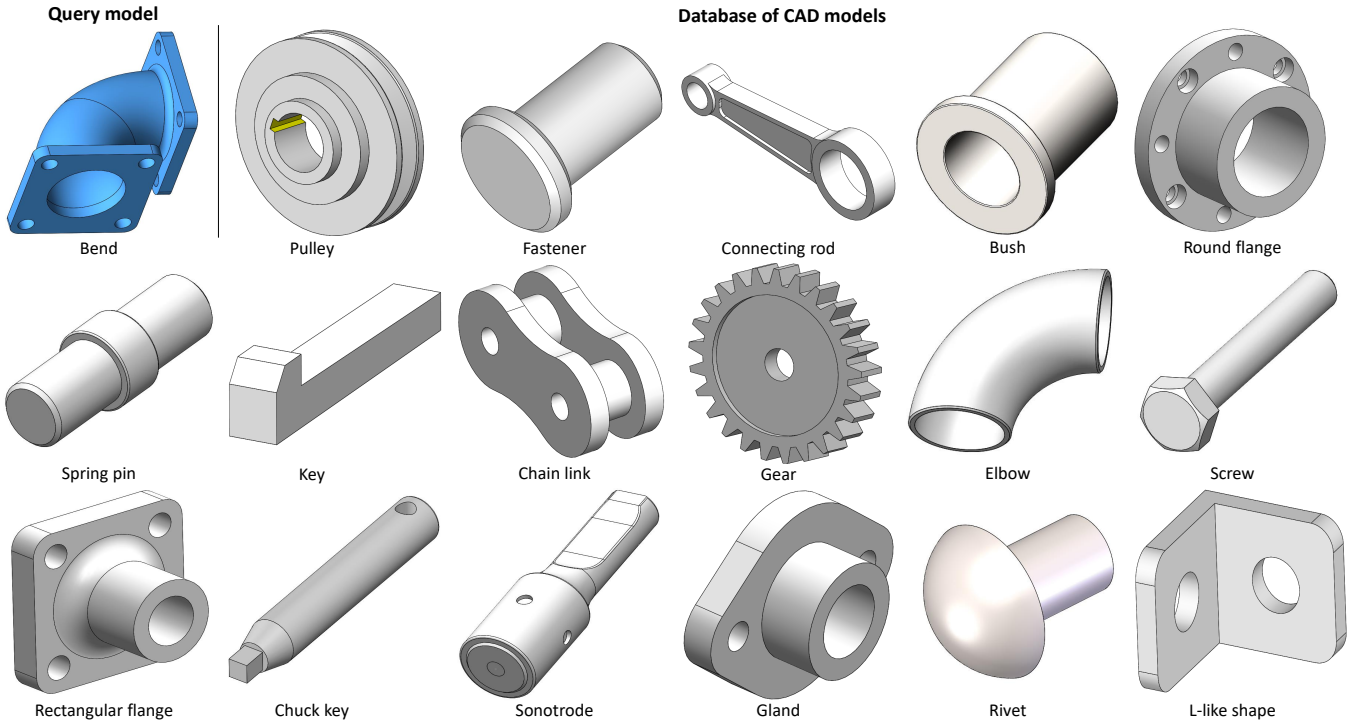


Figure 14: Similarity check between the query model and the models in database.

the bend. Among the models in database, the rectangular flange presents the same pattern and is returned with a similarity measure value of 53% (max = 100). The other models present only a portion of the specified pattern and are correctly returned with smaller similarity measure values.

used in a single group without any weighting strategy. The rest of the setting parameters of configuration C_6 are highlighted in green in Table 4. Figure 13.b shows the fitted bend joint part that has been reconstructed using the best configuration C_6 of the rectangular flange.

CAD Models database	Nodes	Similarity
Pulley	07	14 %
Fastener	02	03 %
Connecting rod	15	23 %
Bush	03	04 %
Spring pin	02	03 %
Key	07	27 %
Chain link	08	18 %
Gear	22	08 %
Elbow	02	04 %
Round flange	15	49 %
Screw	07	20 %
Rectangular flange	15	53 %
Chuck key	06	14 %
Sonotrode	22	46 %
Gland	14	52 %
Rivet	02	04 %
L-like shape	11	35 %

Table 6: Similarity check for the bend part with models in the database.

Having the highest similarity measure for the bend joint, the best configuration of the rectangular flange, i.e. C_6 is used for the fitting of the bend joint. The bend joint whose shape is controlled by 8 parameters is prearranged into the virtually-generated point cloud to initialize the fitting process (Fig. 13.a). Here, the control parameters p_k for the bend are

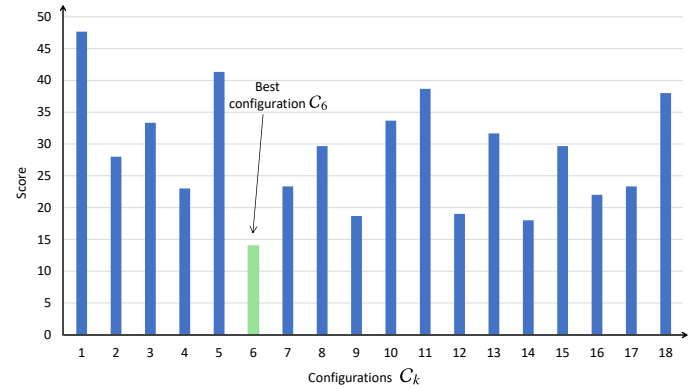


Figure 15: Resulting scores for the 18 configurations used for the SA-based fitting of bend joint.

To evaluate and validate the assumption that best configurations can be used for similar parts, the bend joint is then fitted using the DoEs strategy to identify its best configuration as previously done for the other CAD templates of the database. The same 18 configurations were tested for the bend joint and repeated three times to take the average of the final obtained scores. The resulting scores for all the configurations are shown in Figure 15. The best configuration for the bend joint is also the same as for the rectangular flange, i.e. configuration C_6 . This

validates the idea that new models to be reconstructed can make use of the best configuration of the similar CAD models to produce good quality fitting.

6. Conclusion

This paper has explored the possibility to tune the setting parameters of the SA-based fitting algorithm used for fitting parametrized CAD models locally and globally in the point cloud of a digitized part or an assembly. A framework integrating sensitivity analysis of the parameters has been proposed to understand how the shape driving parameters of the CAD models are modified by the SA algorithm and how such an approach helps to group the CAD model's parameters to treat batches having comparable sensitivities. Using the DoEs approach, the framework also details the steps for the identification of the best configuration of setting parameters for the SA algorithm that will enhance the quality of the fitting approach. A framework including the generation of a database of CAD models stored with their best configurations has also been presented. Such a database of CAD models will help the users to make use of those stored best configurations for the new parts to be fitted by performing a similarity check between the new models and the models in the database.

At the moment, the created database is rather small and only contains few 3D CAD parts. However, the overall framework has been validated. For future perspectives, the database can be enlarged to also contain parametric models and their best configurations for the fitting of 2D sketches and assemblies. Sensitivity analysis performed at the start of the optimization can be integrated into the SA algorithm to track the sensitivities of parameters all along the optimization. Keeping the track of the evolution of sensitivities, the system will automatically handle the less sensitive parameters.

References

- [1] Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 2017;6:1–10.
- [2] Falcidieno, B, Giannini, F, Léon, JC, Pernot, JP. Processing free form objects within a product development process framework. *Advances in Computers and Information in Engineering Research* 2014;;317–344.
- [3] Shah, GA, Polette, A, Pernot, JP, Giannini, F, Monti, M. Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies. *Engineering with Computers* 2020;;1–19.
- [4] Buonamici, F, Carfagni, M, Furferi, R, Governi, L, Lapini, A, Volpe, Y. Reverse engineering of mechanical parts: A template-based approach. *Journal of Computational Design and Engineering* 2018;5(2):145–159. doi:<https://doi.org/10.1016/j.jcde.2017.11.009>.
- [5] Kirkpatrick, S, Gelatt, C, Vecchi, M. Optimization by simulated annealing. IBM research report rc 9355, acts of ptrc summer annual meeting, 1982. 1982.
- [6] Hutter, F, Hoos, HH, Leyton-Brown, K, Murphy, K. Time-bounded sequential parameter optimization. In: *International Conference on Learning and Intelligent Optimization*. Springer; 2010, p. 281–298.
- [7] Gunawan, A, Lau, HC, et al. Fine-tuning algorithm parameters using the design of experiments approach. In: *International Conference on Learning and Intelligent Optimization*. Springer; 2011, p. 278–292.
- [8] Iooss, B, Lemaître, P. A review on global sensitivity analysis methods. In: *Uncertainty management in simulation-optimization of complex systems*. Springer; 2015, p. 101–122.
- [9] Hamby, DM. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment* 1994;32(2):135–154.
- [10] Jin, Y, Meng, X, Ziyou, G. Sensitivity analysis of simulated annealing for continuous network design problems. *Journal of transportation systems engineering and information technology* 2009;9(3):64–70.
- [11] Gamboa, F, Janon, A, Klein, T, Lagnoux, A, et al. Sensitivity analysis for multidimensional and functional outputs. *Electronic Journal of Statistics* 2013;8(1).
- [12] Spagnol, A, Riche, RL, Veiga, SD. Global sensitivity analysis for optimization with variable selection. *SIAM/ASA Journal on Uncertainty Quantification* 2019;7(2):417–443.
- [13] Lamboni, M, Monod, H, Makowski, D. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety* 2011;96(4):450–459.
- [14] Jin, Y, Meng, X, Ziyou, G. Sensitivity analysis of simulated annealing for continuous network design problems. *Journal of transportation systems engineering and information technology* 2009;9(3):64–70.
- [15] Zhan, Sh, Lin, J, Zhang, Zj, Zhong, Yw. List-based simulated annealing algorithm for traveling salesman problem. *Computational intelligence and neuroscience* 2016;2016.
- [16] Bellio, R, Ceschia, S, Di Gaspero, L, Schaerf, A, Urli, T. Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers & Operations Research* 2016;65:83–92.
- [17] Atiqullah, MM, Rao, S. Tuned annealing for optimization. In: *International Conference on Computational Science*. Springer; 2001, p. 669–679.
- [18] Giannini, F, Lupinetti, K, Monti, M. Identification of similar and complementary subparts in B-rep mechanical models. *Journal of Computing and Information Science in Engineering* 2017;17(4).
- [19] Montlahuc, J, Shah, GA, Polette, A, Pernot, JP. As-scanned point clouds generation for virtual reverse engineering of cad assembly models. *Computer-Aided Design and Applications* 2018;;1–11.
- [20] Introduction to taguchi method, chapter 2. 2020. URL: <http://www.ecs.umass.edu/mie/labs/mda/fea/sankar/chap2.html>; accessed: 2020-09-23.
- [21] Roy, RK. A primer on the Taguchi method. *Society of Manufacturing Engineers*; 2010.
- [22] Gonzales, GV, Dos Santos, ED, Emmendorfer, LR, Isoldi, LA, Rocha, LAO, Estrada, E. A comparative study of simulated annealing with different cooling schedules for geometric optimization of a heat transfer problem according to constructal design. *Sci Plena* 2015;11(8):11.
- [23] Ingber, L. Adaptive simulated annealing (asa): Lessons learned. *Control and Cybernetics* 1996;25(1):32–54. Cited By 364.

3.2 Conclusion

This chapter has presented the possibility to identify and store the best configuration of parameters for the Simulated Annealing algorithm that would produce the optimal reconstruction results using the Simulated Annealing-based Reverse Engineering technique. Various examples related to local and global fitting are tested. The impact of the choice of parameters for the Simulated Annealing algorithm has been presented using DoEs to evaluate the quality of fitting on the basis of three criteria: the overall energy of the system at the end of optimization; the total number of iterations taken by Simulated Annealing for the fitting; and the final sum of absolute deviations of parameters of the reconstructed CAD model. The resulting experiments from DoEs are scored to identify the best configuration of parameters for the Simulated Annealing-based reconstruction process. A new framework for the creation of a small database of CAD models stored with their best identify configurations of parameters has also been presented. Such a database of CAD models will help the users to make use of those stored best configurations for the new parts to be fitted by performing a similarity check between the new models and the models in the database. Now that fitting approach has been tuned, it can be used within the new RE framework discussed in the next chapter.

CHAPTER



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

Contents

4.1	Introduction	70
4.2	Conclusion	85

4.1 Introduction

After successfully devising strategies for as-scanned point cloud generation and fitting techniques in the previous chapters respectively, a new Reverse Engineering framework is introduced in this chapter (highlighted in green in Fig. 1). Fitting remains the core of this multi-criteria new Reverse Engineering technique that allows the reconstruction of editable CAD models of parts and assemblies. Unlike the part-by-part reconstruction process detail in the previous chapter (Chapter 2), the reconstruction process starts from roughly sketched and parameterized 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 reconstructed geometries. So, in this chapter, not only we can work on CAD parts and assemblies but also on 2D sketches used to reconstruct the parts starting from scratch. The proposed method in this chapter is supported by a two-level filtering technique able to capture and manage the boundaries of the geometries inside the overall point cloud in order to allow for local fitting and interface detection. This was not the case in the previous chapter where the filtering was performed using a simple distance criterion inside the segmentation loops.

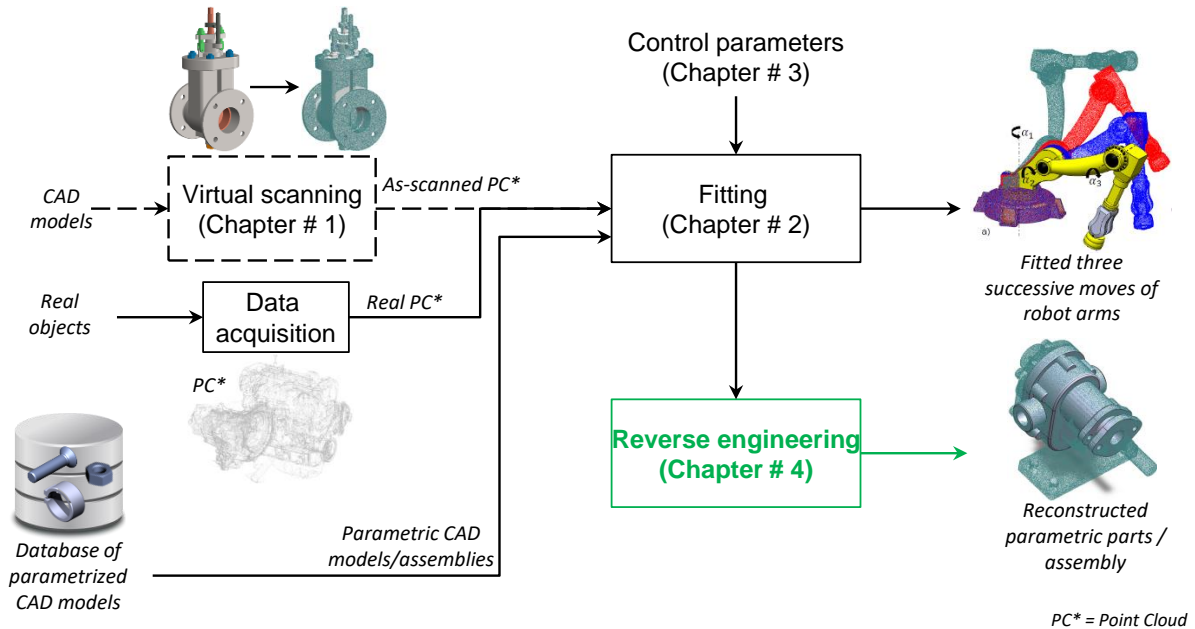


Figure 1 – Framework for the new reverse engineering technique for efficient reconstruction of the mechanical parts/assembly models.

The method is much faster as compared to the traditional reverse engineering where engineers face

4.1. INTRODUCTION

a time-consuming patch-by-patch fitting strategy during which they have to face many issues: pre-processing of the data, segmentation of point clouds, decomposition in patches, fitting of primitives, trimming and stitching of the resulting surfaces. Following those approaches, manifold B-Rep models are obtained but cannot be easily modified as they are considered dead models which do not rely on real building trees. This is however not the case in our RE framework wherein the resulting CAD models are parametrized and can thus be easily modified in the forth coming steps of the PDP. The efficiency of the new Reverse Engineering framework has been validated on several examples including single and multiple parts for both global and local fitting cases. Experiments have also been performed on both real and as-scanned point clouds.

4.1. INTRODUCTION

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

Abstract

This paper introduces a novel reverse engineering 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 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 reconstructed geometries. The coherence of the CAD models is maintained by a CAD modeler that performs the updates and satisfies the geometric constraints as the fitting process goes on. 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 for local fitting and interfaces detection. 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.

Keywords: Reverse Engineering, CAD assembly models reconstruction, point cloud filtering, segmentation, 2D and 3D fitting

1. Introduction

Being able to get access to 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 interest has accelerated due to the rapid evolution of digitization technologies, particularly in the scope of the 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 for what concerns 3D meshes. Indeed, less attention has been paid to the generation or update of CAD models, which are nevertheless often 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 [2]. Quite few 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. 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 (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. The input is a point cloud of a mechanical parts' assembly that has been acquired as a whole. The proposed framework allows for the 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 used as input of a simulated annealing-based fitting algorithm, which minimizes the deviation between the point cloud and the reconstructed 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; (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 interfaces detection. The advantages of the proposed approach are demon-

strated on several industrial examples, and a comparative study with commercial softwares 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 for advanced industrial applications, where backward engineering is used to build a CAD model that is geometrically identical to the existing physical product or system [3]. New low-cost data acquisition devices are emerging to digitize 3D objects with enhanced quality and reduced acquisition times. Many researchers from the field of image processing, computational geometry, and computer graphics have proposed different approaches to reconstruct surfaces from the obtained point cloud [4]. 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 geometric 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 [5]. In fine, the studied literature can be split using three main priors: the 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. Sophisticated surface fitting algorithms are required to generate surfaces that accurately represent the 3D information described within the point cloud [6]. Surface-based reconstruction is a more traditional way of reverse engineering. 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 multiple research works [7, 8, 9, 10, 11]. 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 [12]. Azernikov et al. [13] 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 [14]. 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. [15] 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 [16], 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. Monzspart 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 [17]. 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 [18]. 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. [19] 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. [20] 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. Similarly, Benko et al. devised a method for the simultaneous fitting of multiple curves and surfaces to 3D measured data captured as part of a reverse engineering process, where constraints exist between the parameters of the curves or surfaces [21]. They found a way to fit parameterized objects (e.g. curves, surfaces) by optimizing the parameter's values using least-squares distance and using an initial guess of the values of the parameters to be optimized and fitted in the given reference point cloud. The proposed approach is discussed on very basic geometries like lines, circles, and surfaces to be fitted using geometric constraints. Another approach is adopted by Durupt et al. to obtain a real CAD model

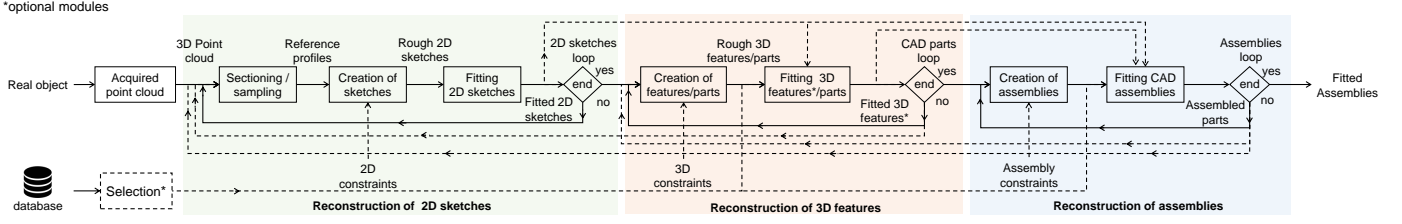


Figure 1: Overall framework for the new reverse engineering technique exploiting a generic multi-dimensional fitting module.

with a tree structure of features called functional and structural skeleton [22]. 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. [23] 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, 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 they 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 CAD models or assemblies 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 Figure 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) reconstruc-

tion 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}^κ 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^κ can be a sketch, a CAD model or an assembly. In the following sections, the three modules are first introduced separately, and section 3.4 discusses configurations where \mathbb{G}^κ 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^κ . Generally speaking, each sketch \mathcal{S}_i^κ 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 any 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).

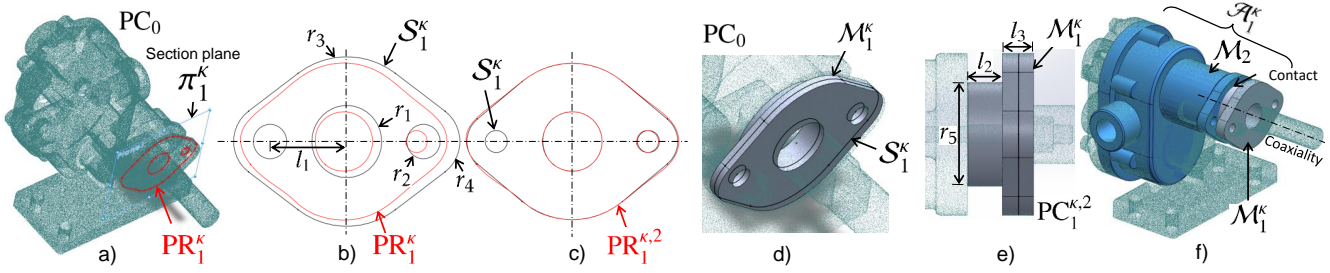


Figure 2: Fitting a parameterized 2D sketch and 3D features in the point cloud of a digitized pump assembly: (a) sectioning with plane π_1^κ to extract the reference profile PR_1^κ in the initial point cloud PC_0 ; (b) drafting a rough parametric sketch S_1^κ in the vicinity of the reference profile PR_1^κ ; (c) final 2D parametric sketch S_1^κ fitted in the filtered profile $PR_1^{\kappa,2}$; (d) draft CAD model M_1^κ extruded from the fitted 2D parametric sketch S_1^κ with an estimated extrusion length; (e) 3D CAD model M_1^κ fitted in the filtered point cloud $PC_1^{\kappa,2}$; (f) fitted assembly \mathcal{A}_1^κ containing two models M_1^κ and M_2^κ assembled together with coaxiality and contact constraints.

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 $\mathbb{G}^\kappa = \{M_i^\kappa, i \in [1..N_m^\kappa]\}$ 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 M_i^κ 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}^κ 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 M_i^κ (section 4.3.3). The consistency of the CAD models is ensured by the CAD modeler in charge of the updates. This process can be repeated for each part to be recovered (Fig. 2.f), either sequentially or 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. 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 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. Then, the fitting algorithm acts on the parameters (e.g. length, angle, radius, distance) while satisfying all the constraints (e.g. parallelism, coaxiality) at different levels (2D sketch, part, assembly) and at the same time. 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}^κ of N_g^κ geometries \mathcal{G}_i^κ in a reference point cloud PC_0 at step κ of the reverse engineering process. This module is generic and works on multi-dimensional geometries, i.e. 2D sketches, CAD models 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).

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 Figure 2.d. The pre-arrangement is always necessary when the reference CAD model is taken from an existing database.

This initialization step also acts on PC_0 while 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 $PC_i^{\kappa}(0)$

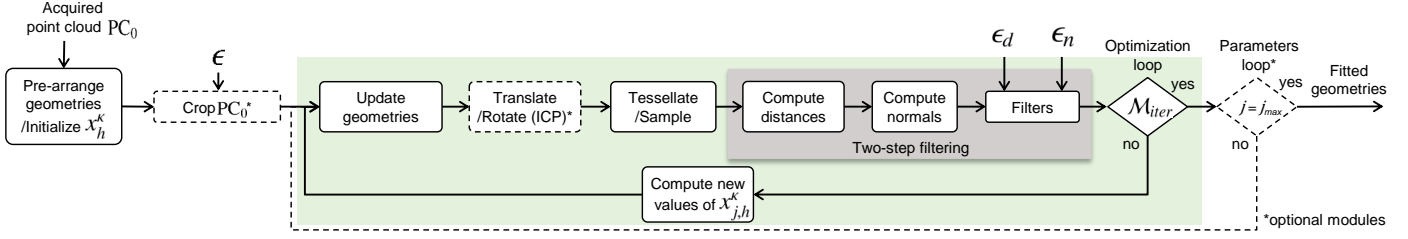


Figure 3: Generic multi-dimensional fitting process handling parameters and constraints at the level of sketches, parts and assemblies.

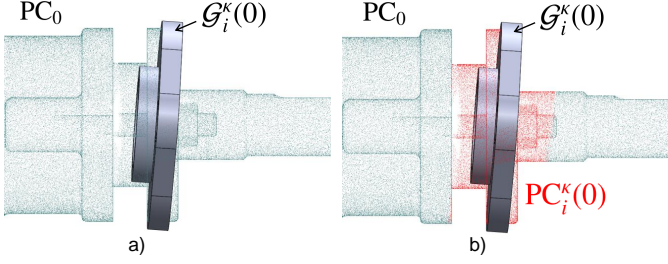


Figure 4: Pre-arrangement of a single CAD model $\mathcal{G}_i^k(0)$ in a reference point cloud PC_0 (a), and initial coarse cropping of PC_0 using a distance threshold to get the cropped point cloud $PC_i^k(0)$ (b).

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 geometry's 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^k can optionally be clustered in several groups G_j^k , 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 overfitting of small features (e.g. fillets and chamfers). Parameters of group G_j^k are denoted $p_{j,h}^k$, $h \in \{1, \dots, N_{p_j}^k\}$ and are assigned to the numerical variables for the optimization loop, i.e. $x_{j,h}^k = p_{j,h}^k$. 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^k groups the parameters of the assembly structure as well as the ones driving the structural features (e.g. pockets, revolutions), G_2^k gathers the parameters of the detail features (e.g. holes, ribs) and G_3^k 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, when referring to those groups, the lower index i of each geometry \mathcal{G}_i^k is not used.

It is important to understand that it is up to the user to decide which 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 Figure 2. Overall, it is controlled by 8 control parameters ($N_p^k = 8$): 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 ($p_{j,h}^k$) are divided into two groups as follows (upper index κ removed for sake of clarity):

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

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

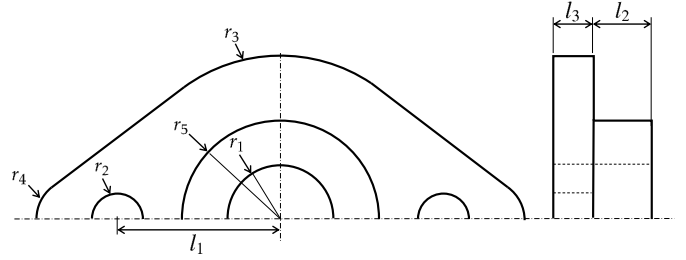


Figure 5: Definition of the parameters controlling the gland of a pump: 5 parameters for the 2D sketch and 3 parameters for the 3D features.

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 $\mathbb{G}^k(0)$ of pre-arranged geometries $\mathcal{G}_i^k(0)$, the algorithm iterates on the values $x_{j,h}^k(t)$ belonging to the j -th group G_j^k , thus creating an evolution of the N_g^k geometries $\mathcal{G}_i^k(t)$ until reaching a best fit of all the geometries in the filtered point clouds $PC_i^k(t)$. Actually, this can be formulated as a minimization problem:

$$\min_{\substack{x_{j,h}^k \in \mathcal{D}_{j,h}^k \\ h \in [1..N_{p_j}^k]}} E_j^k(x_{j,1}^k, \dots, x_{j,N_{p_j}^k}^k) = \sum_{i=1}^{N_g^k} d(PC_i^k, \mathcal{G}_i^k) \quad (2)$$

where $x_{j,h}^k$ are the variables, $\mathcal{D}_{j,h}^k$ their definition domains, and E_j^k the energy function characterizing the overall deviation for the complete set \mathbb{G}^k at the j -th step of the parameters loop. The filtering technique is explained in section 4.3.3, and the way

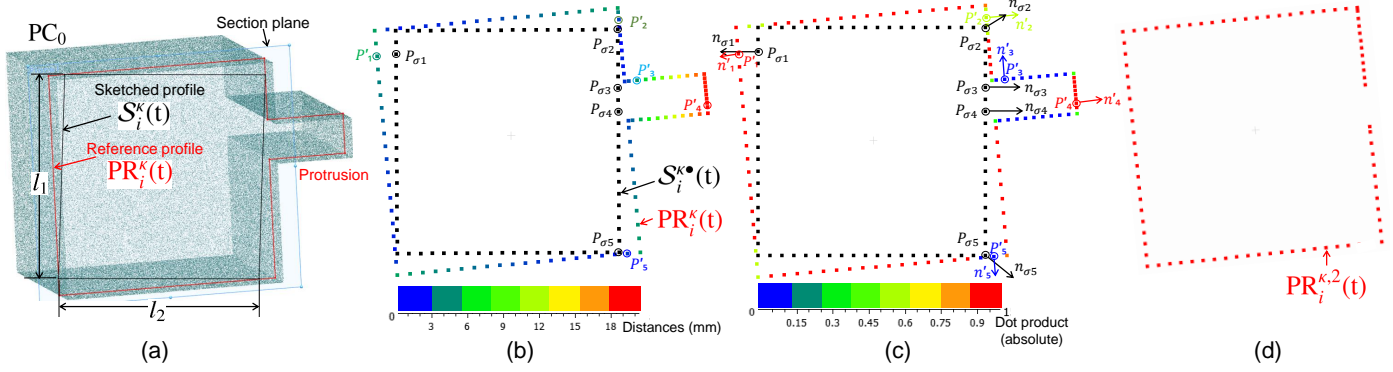


Figure 6: The two-level filtering technique applied on a square-shaped 2D sketch: (a) parameterized sketch $S_i^k(t)$ close to the reference profile $PR_i^k(t)$ with a protrusion not to be considered during the fitting; (b) points of $S_i^k(t)$ and $PR_i^k(t)$ after sampling, with colors showing the 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 $PR_i^{k,2}(t)$ defining the point cloud used for the computation of the energy.

distances between the filtered $PC_i^{k,2}$ and the geometry \mathcal{G}_i^k 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 $\mathcal{G}_i^k \in \mathbb{G}^k$ in order to fit the filtered point clouds $PC_i^{k,2}$, the reconstruction process requires a metaheuristic 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 that performs a global stochastic search by evolving towards a local search along the time. 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 left to 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 [24, 25].

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_g^k geometries involved at the k -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 $\mathcal{G}_i^k(t)$ is thus updated by the CAD modeler according to the evolution of the parameters values $x_{j,h}^k(t)$. Here, updates can appear at the level of the sketches, parts and assemblies. Then, each geometry is translated and rotated using an ICP algorithm [26] that finds the best fit rigid body transformation between the geometry and its associated $PC_i^k(t)$. 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 $\mathcal{G}_i^{k,\Delta}(t)$ after the tessellation and $\mathcal{G}_i^{k,*}(t)$ after the sampling. The sampling is performed so as to keep same the density of both the sampled geometry $\mathcal{G}_i^{k,*}(t)$ and its corresponding cropped point cloud $PC_i^k(t)$. In case of a 2D sketch, the cropped point cloud is a reference profile $PR_i^k(t)$ 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 $\mathcal{G}_i^k(t)$. 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. Figure 6 illustrates this filtering process on the example of a square-shaped 2D sketch $S_i^k(t)$ to be fitted to a reference profile $PR_i^k(t)$ presenting a protrusion not to be considered during the local fitting. At a current 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 $\mathcal{G}_i^{k,*}(t)$ and the ones of the reference point cloud $PC_i^k(t)$, respectively $S_i^{k,*}(t)$ and $PR_i^k(t)$ on the 2D example of Figure 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:

$$\begin{aligned} \|P'_k P_{\sigma(k)}\| &< \epsilon_d \\ |n'_k \cdot n_{\sigma(k)}| &< \epsilon_n \end{aligned} \quad (3)$$

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 $PC_i^{\kappa,1}(t)$ 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 $PC_i^{\kappa,2}(t)$ 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 Figure 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 equation (2) is computed using the following function:

$$d(PC_i^{\kappa,2}(t), \mathcal{G}_i^{\kappa}(t)) = \sum_{\tau=1}^{n_i^{\kappa}(t)} d^2(\mathcal{G}_i^{\kappa}(t)[\tau], PC_i^{\kappa,2}(t)) \quad (4)$$

where $n_i^{\kappa}(t)$ is the number of points in $\mathcal{G}_i^{\kappa}(t)$. The function $d(\text{point}, \text{cloud})$ returns the the closest distance between the point and 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 computes new values $x_{j,h}^{\kappa}(t+1)$ defining a new set $\mathcal{G}^{\kappa}(t+1)$ 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 tessellate 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 [27], which uses the Hidden Point Removal (HPR) algorithm from 12 viewpoints [28]. Before generating the point cloud, the CAD assembly was simplified while removing and remounting some 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:

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

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

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
	G_3	r_9	13	11	11.983	-0.9830	0.0894
		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)	<i>fillet</i>	2	5	6.7520	-1.750	0.3500

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

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

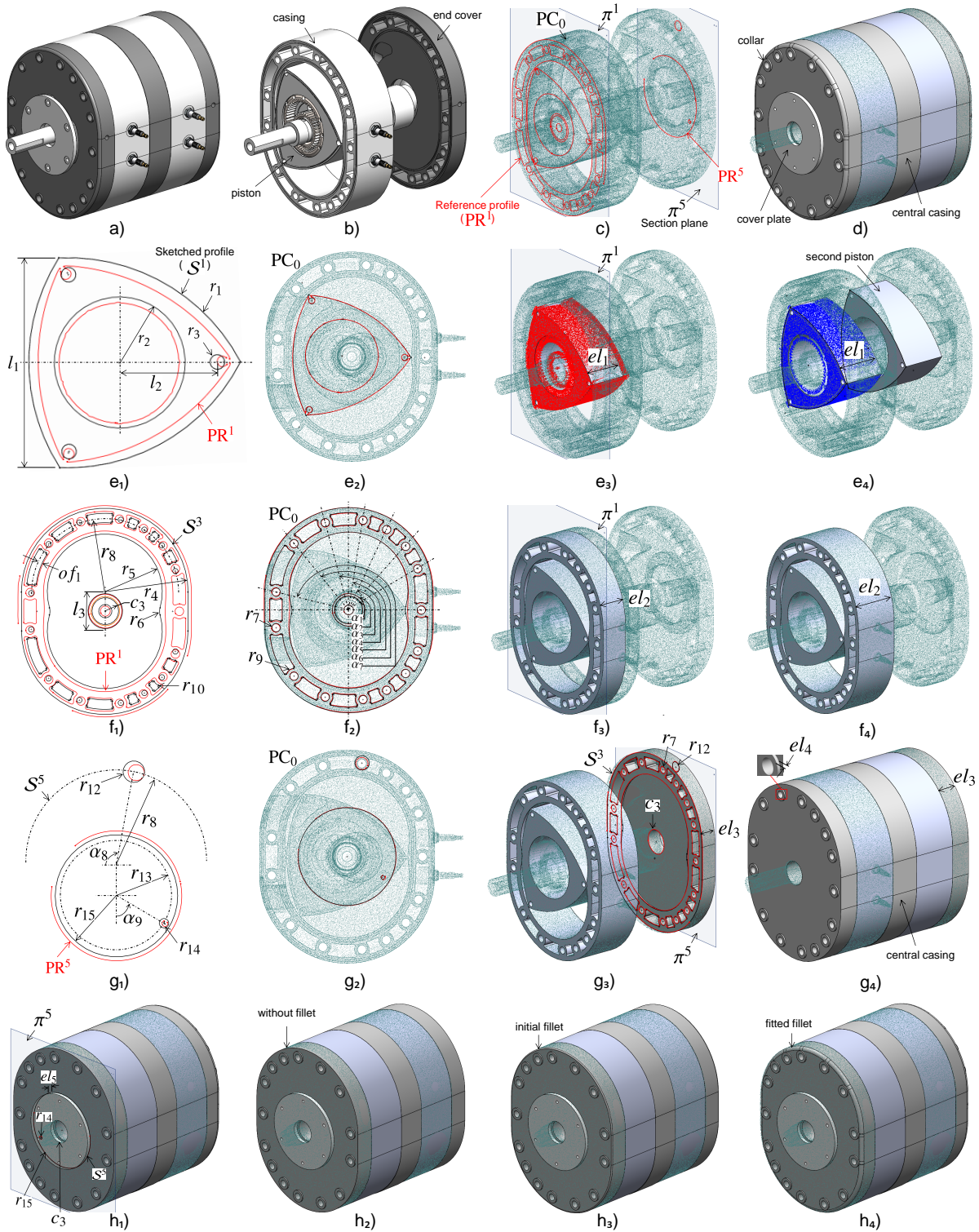


Figure 7: Successive steps of the reverse engineering of a Wankel engine with a point cloud generated from an as-scanned virtual scanning.

($\kappa \in \{5, 6, 7\}$), 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.e₁ to 7.e₄): 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 S^1 is drafted around the reference profile related to the piston only (Fig. 7.e₁). ICP is also used to handle the align-

ment 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.e₂). 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.e₃). 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.e₃ 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.e₄). 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.e₄).

The next component to be reversed is the casing following a two-step procedure (Figures 7.f₁ to 7.f₄): 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.f₁ and 7.f₂) and clustered in 3 groups according to the level of detail they refer to (Table 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. The stochastic nature of the SA algorithm is clearly visible. Such an algorithm 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.f₃ and 7.f₄).

Then, the end cover is reverse engineered in three steps (Figures 7.g₁ to 7.g₄): 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.g₁ and 7.g₂). 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.g₃). The collar is then recovered while optimizing its extrusion length el_4 in step $\kappa = 7$ and the resulting feature is repeated and aligned 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.g₄).

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.h₁ and 7.h₂).

Finally, a fillet is added to the end cover, and its radius is optimized at step $\kappa = 9$ (Figures 7.h₃ and 7.h₄). All the optimized

values are reported in Table 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.

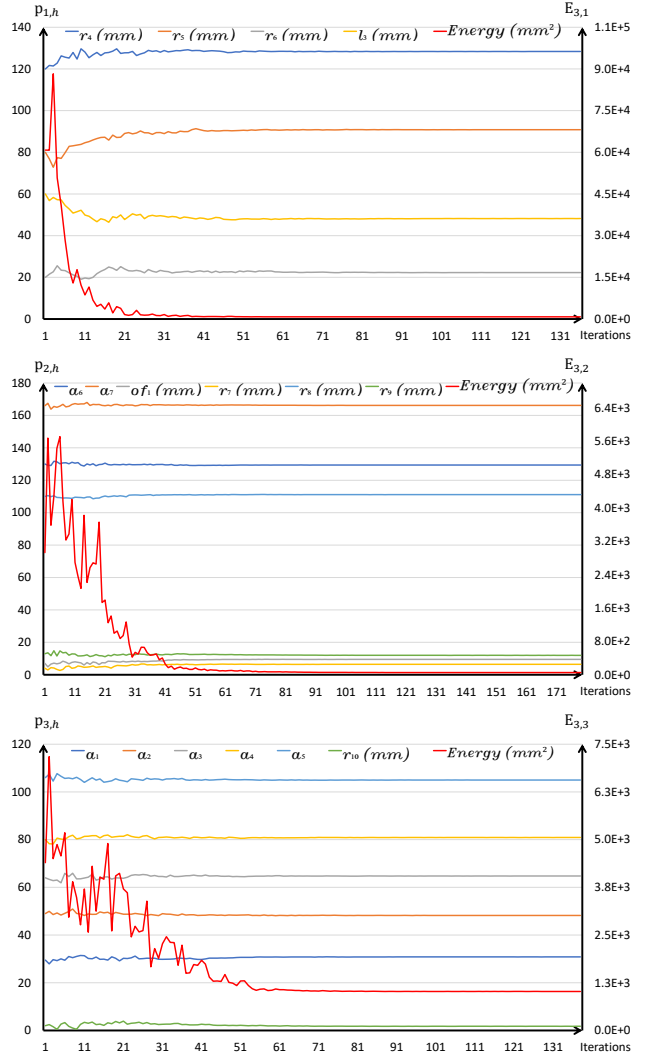


Figure 8: Simulated annealing evolution curves for the three parameters groups to reconstruct the parametric sketch of the casing part.

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 it prior to its digitalization. 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 (Table 2).

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_1	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

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

Due to space limitation, only the reconstruction of the 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.e₁ to 9.e₄): 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.e₁ and 9.e₂). The resulting values p_h^F after the optimization are

listed in Table 2. Then, at step $\kappa = 2$, the extrusion length el_1 is optimized using SA algorithm from a user-specified coarse value (Figures 9.e₃ and 9.e₄). 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.f₁ to 9.f₄): 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.f₁ and 9.f₂). Here, one can notice that 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.f₃ and 9.f₄). 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.g₁ and 9.g₂). 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.g₃ and 9.g₄). As for all the other steps, the fitting is performed using the two-level filtering technique to be able 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 one parameter br_1 that is directly fitted using SA algorithm (Figures 9.h₁ and 9.h₂). The screw head is generated while optimizing its extrusion length el_4 (Fig. 9.h₃), and the bottom part of the screw is directly reconstructed using an extrusion of length $el_5 = el_2 + el_3$ (Fig. 9.h₄). During the fitting, the screw is constrained to stay on the circle of radius r_2 . Of course, screws being standard parts, the obtained values can be aligned on normalized values available from a catalog. As for the thickness of the cylinder, one can also notice that the point cloud being obtained without disassembling the parts, the length of the screw that is screwed inside the central casing cannot be directly retrieved and should be further 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.i₁). 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 used for fitting, to adjust the parameters values already fitted on the right end side. Here, all the duplicated geometries

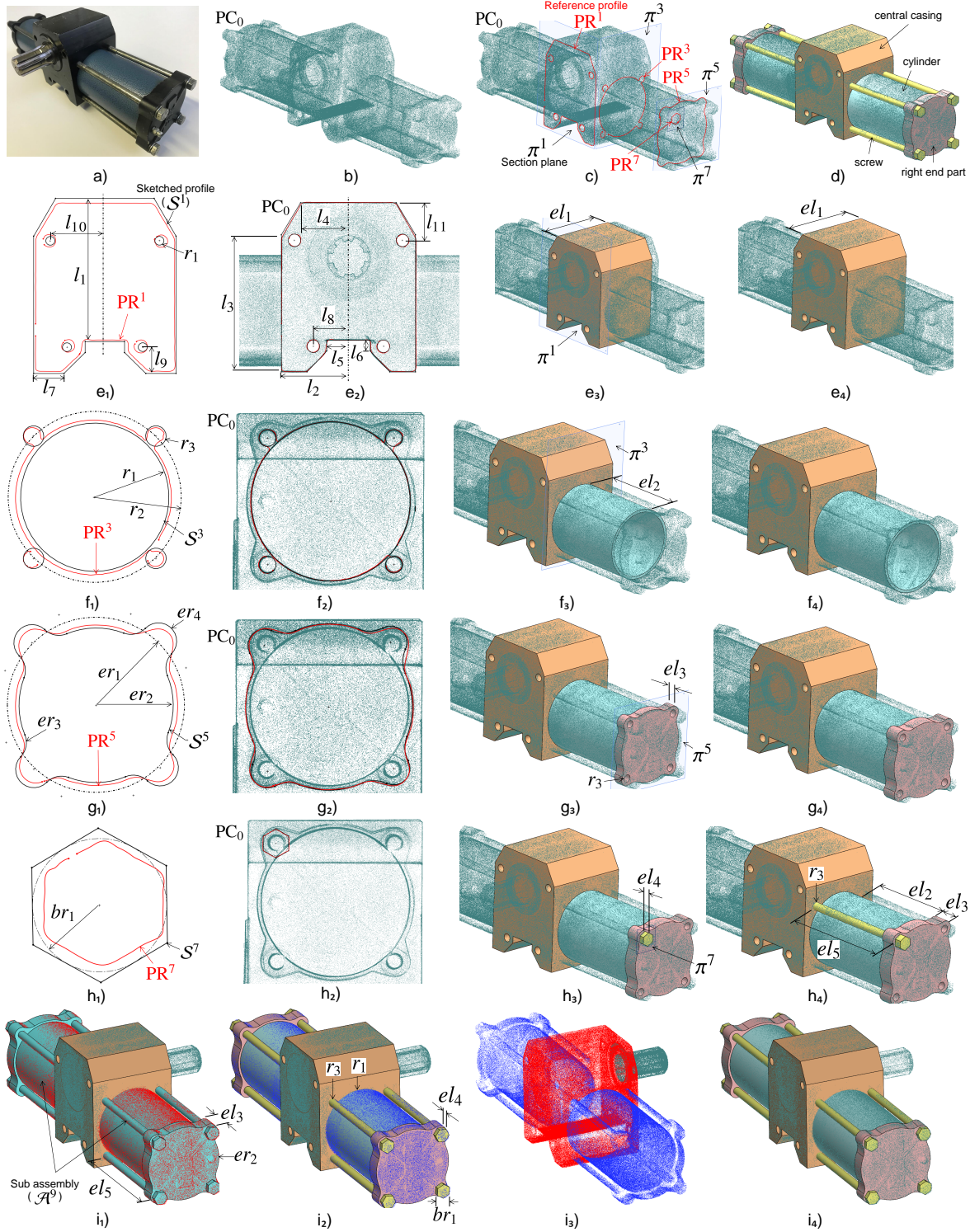


Figure 9: Successive steps of the reconstruction of a hydraulic rotary actuator based on a point cloud obtained using a laser scanner without disassembling the parts.

take part to the optimization process and the only parameters that are further tuned are listed in Table 2. 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 $r_2 = r_1$. Figure 9.i4 shows the result of the final fitted assembly, and figure 9.i3 the segmented point cloud distinguishing points belonging to the fitted sub-assembly (blue) from the ones of the central casing (red).

6. Conclusion and future works

This paper has introduced a novel framework for the reconstruction of parametric sketches, CAD models and assemblies from point clouds. The proposed approach bypasses the traditional reverse engineering technique where surfaces are reconstructed patch-by-patch and used 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, which can for instance be rounded if necessary. 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. coaxiality 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 [29].

References

- [1] Lu, Y. Industry 4.0: a survey on technologies, applications and open research issues. *Jou of Industrial Information Integration* 2017;6:1–10.
- [2] Falcidieno, B, Giannini, F, Léon, JC, Pernot, JP. Processing free form objects within a product development process framework. *Advances in Computers and Information in Engineering Research* 2014;;317–344.
- [3] Dúbravčík, M, Kender, Š. Application of reverse engineering techniques in mechanics system services. *Procedia Engineering* 2012;48:96–104.
- [4] De Toledo, R, Levy, B, Paul, JC. Reverse engineering for industrial-environment CAD models. In: *International Symposium on Tools and Methods of Competitive Engineering, TMCE 2008*. Izmir, Turkey; 2008,URL: <https://hal.inria.fr/inria-00202465>.
- [5] Anwer, N, Mathieu, L. From reverse engineering to shape engineering in mechanical design. *CIRP Annals* 2016;65(1):165–168.
- [6] Shabani, B, Pandilov, Z. Analyzing and application of reverse engineering for design and development of mechanical parts. *Mechanical Engineering - Scientific Journal* 2017;35(2):89–96.
- [7] Petitjean, S. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys (CSUR)* 2002;34(2):211–262.
- [8] Lukács, G, Marshall, A, Martin, R. Geometric least-squares fitting of spheres, cylinders, cones and tori. *RECCAD, Deliverable Document 2 and 3, COPERNICUS project 1997*;(1068).
- [9] Werghi, N, Fisher, R, Ashbrook, A, Robertson, C. Faithful recovering of quadric surfaces from 3d range data. In: *2nd Int. Conf. on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*. 1999, p. 280–289.
- [10] Satish, A, Rambabu, K, Ramji, M. Part modelling with reverse engineering. *Int J of Eng Resch & Tech* 2013;2:486–94.
- [11] Mohaghegh, K, Sadeghi, M, Abdullah, A. Reverse engineering of turbine blades based on design intent. *The International Journal of Advanced Manufacturing Technology* 2007;32(9-10):1009–1020.
- [12] Kumar, A, Jain, P, Pathak, P. Reverse engineering in product manufacturing: an overview. *DAAAM Int Scientific Book* 2013;;665–678.
- [13] Azernikov, S, Fischer, A. Efficient surface reconstruction method for distributed CAD. *Computer-Aided Design* 2004;36(9):799–808.
- [14] Mitra, N, Wand, M, Zhang, HR, Cohen-Or, D, Kim, V, Huang, QX. Struct-aware shape process. *SIGGRAPH Asia 2013 Courses* 2013;;1–20.
- [15] Li, Y, Wu, X, Chrysathou, Y, Sharf, A, Cohen-Or, D, Mitra, NJ. Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics* 2011;30(4):1–12.
- [16] Schnabel, R, Wahl, R, Klein, R. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum* 2007;26(2):214–226.
- [17] Monszpart, A, Mellado, N, Brostow, GJ, Mitra, NJ. Rapter : Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans Graph* 2015;34(4):1–12.
- [18] Bey, A, Chaine, R, Marc, R, Thibault, G, Akkouche, S. Reconstruction of consistent 3d CAD models from point cloud data using a priori CAD models. In: *ISPRS workshop on laser scanning; vol. 1*. 2011,.
- [19] Buonamici, F, Carfagni, M, Furfieri, R, Governi, L, Lapini, A, Volpe, Y. Reverse engineering of mechanical parts: A template-based approach. *Journal of Computational Design and Engineering* 2018;5(2):145–159.
- [20] Erdős, G, Nakano, T, Váncza, J. Adapting cad models of complex engineering objects to measured point cloud data. *CIRP Annals* 2014;63(1):157–160.
- [21] Benko, P, Kós, G, Várady, T, Andor, L, Martin, R. Constrained fitting in reverse engineering. *Computer Aided Geometric Design* 2002;19(3):173–205.
- [22] Durupt, A, Remy, S, Ducellier, G. Reverse engineering of a piston using knowledge based reverse engineering approach. In: *Global Product Development*. Springer; 2011, p. 683–690.
- [23] Thompson, WB, Owen, JC, Germain, HdS, Stark, SR, Henderson, TC. Feature-based reverse engineering of mechanical parts. *IEEE Transactions on robotics and automation* 1999;15(1):57–66.
- [24] Gouaty, G, Fang, L, Michelucci, D, Daniel, M, Pernot, JP, Raffin, R, et al. Variational geometric modeling with black box constraints and dags. *Computer-Aided Design* 2016;75:1–12.
- [25] Pernot, JP, Michelucci, D, Daniel, M, Fofou, S. Towards a better integration of modelers and black box constraint solvers within the product design process. *Annals of Mathematics and Artificial Intelligence* 2019;85(2):147–173.
- [26] Besl, PJ, McKay, ND. A method for registration of 3-d shapes. *IEEE Trans Pattern Anal Mach Intell* 1992;14(2):239–256.
- [27] Montlahuc, J, Shah, GA, Polette, A, Pernot, JP. As-scanned point clouds generation for virtual reverse engineering of cad assembly models. *Computer-Aided Design and Applications* 2018;;1–11.
- [28] Katz, S, Tal, A, Basri, R. Direct visibility of point sets. *ACM Transactions on Graphics* 2007;26(3):1–11.
- [29] Danglade, F, Pernot, JP, Véron, P, Fine, L. A priori evaluation of simulation models preparation processes using artificial intelligence techniques. *Computers in Industry* 2017;91:45–61.

4.2 Conclusion

This chapter has presented a new Reverse Engineering framework that can reconstruct 2D parametric sketches along with 3D single and assembly models reconstruction. This is especially helpful for cases where roughly drafted parametric CAD models are not available for the reconstruction process. The proposed reconstruction process can start by sketching a rough 2D parametric sketch, which will be fitted to an extracted contour profile obtained from the reference point cloud. To avoid the basic segmentation loops for the cases of local fitting (Chapter 2), a new two-level filtering segmentation technique is presented to automatically segment the point cloud during the reconstruction process. The new Reverse Engineering technique bypasses the traditional patch-by-patch surfaces reconstruction process to produce editable CAD models that can later be used in the stages of Product Development Processes. Both local and global fitting can be performed, considering geometries either one-by-one or all together to further optimize the fitting afterward. This is particularly interesting to avoid disassembling parts to be scanned individually. The method is modular, where each module performs specific task to support the new Reverse Engineering framework. For example, 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. coaxiality between axes, contact between faces, relationships between parameters). The efficiency of the new Reverse Engineering framework has been tested on both real scanned and virtually generated as-scanned point clouds.

4.2. CONCLUSION

Conclusion & perspectives

Conclusion

The ability to reconstruct or update 3D representations of existing products and systems has become mainstream to support the creation and exploitation of digital twins in the context of Industry 4.0. The relative ease of access to more or less sophisticated 3D acquisition devices has certainly accelerated the demand while extending the range of possible applications. However, the acquired point cloud treatment and the reverse engineering of CAD models are not yet fully automatized. The resulting models are not fully exploitable for CAD software and the existing patch-by-patch surface reconstruction does not fulfill the requirements of the Industry 4.0 needs. Most of the existing tools for reverse engineering ends up with non-editable dead models in which the structure of the product or system does not change, and only the position and orientation of its constitutive parts is to be updated. Thus, the update problem can be transformed in a fitting problem for which the optimal parameter values of CAD models have to be found to minimize the deviation between the digitized point clouds and the digital twins. Reverse engineering of editable CAD geometries, which can later be used and modified in the product development processes (PDP), remains the main focus of the researchers especially after the availability of low-cost data acquisition devices. To overcome these limitations, the general objective of this thesis was the proposition of a new Reverse Engineering technique for the reconstruction of editable CAD models of mechanical parts' assemblies.

The point clouds used for the fitting and new Reverse Engineering technique were either obtained from a real laser scanner (ROMER Absolute Arm 7520 SI, 7 axis and 2m volume, absolute encoders, RS1 laser sensor 30000 pts/s, volumetric accuracy of 61 μ m) or through the as-scanned point cloud generation technique in which point clouds are generated from the existing CAD models and assemblies incorporating various artifacts that would appear if they were scanned with real scanners. The as-scanned point cloud generation technique is controlled by several parameters which values can be used to insert artifacts commonly encountered when dealing with real acquisition devices and parameters used for generating these results can be fine-tuned with the parameters of scanners for sensitivity and accuracy. The method is very fast when compared to the traditional Reverse Engineering process and it does not require any tedious and time-consuming post-processing steps.

A framework for the new Reverse Engineering technique has been introduced for the reconstruction of parametric sketches, CAD models and assemblies from point clouds. The proposed approach bypasses the traditional reverse engineering technique where surfaces are reconstructed patch-by-patch and used

to create dead CAD models that cannot be exploited later in the PDP. The strengths of the approach lie in the definition and use of a multidimensional 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. The proposed approach is very promising when considering the need to maintain the coherence between a physical system and its digital twin in the scope of the Industry 4.0. It has been tested on several configurations, for which as-scanned point clouds have been generated following an ad hoc virtual scanning approach. Thus, it has been possible to measure the deviations between the parameters of the fitted parts, and the ones of the original parts as they appear in the DMU.

A first attempt to define a tracking approach able to maintain the coherence between a physical system and its digital twin has proved to be promising, and it already gives good results on a simple 3-axes robot. 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. coaxiality 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.

Moreover, the proposed framework for the new Reverse Engineering integrating sensitivity analysis of the parameters has been proposed to understand how the shape driving parameters of the CAD models are modified by the SA algorithm and how such an approach helps to group the CAD model's parameters to treat batches having comparable sensitivities. A framework including the generation of a database of CAD models stored with their best configurations has also been presented. Such a database of CAD models will help the users to make use of those stored best configurations for the new parts to be fitted by performing a similarity check between the new models and the models in the database.

Comparison with commercial software

To show the strengths and weaknesses of our proposed approach, we have compared our proposed work with two commercial software, i.e. CATIA V5 and DesignX. Table 3 highlights some features of our methods in comparison with the two mentioned software.

Traditional software for reverse engineering mostly works with the idea of patch reconstruction that is later trimmed and stitched together to obtain a closed surface geometry. This geometry is then transformed into a volumetric shape. Differently, our method does not require patch-by-patch reconstruction, it does support such capability to perform the patch-by-patch fitting. Patches can be produced from the template of the CAD geometries by offsetting surfaces which are then optimized for fitting using distance constraints. With our method, the CAD template to be reconstructed incorporates features that are generated by the CAD operation, features like fillets and chamfer can be reconstructed even at end of the optimization process if they are not considered in the early stage of the optimization. Features are not recognized in the reconstruction process of the CATIA while DesignX allows the reconstruction of a limited number of features like revolves, extrusion, etc. Amongst the strengths of the proposed method is a reconstruction technique that allows either the reconstruction of a CAD template part-by-part or of a whole assembly. CAD template is reconstructed one after the other or simultaneously to reconstruct an assembly. This capability is missing in the existing traditional RE software like CATIA and DesignX. Constraint satisfaction also plays an important role during the reconstruction process. For instance, in traditional RE, if constraints like concentricity, parallelism, perpendicularity, etc. are not maintained properly during the reconstruction process, it is difficult to address them at the end of the reconstruction process. In our method, it is up to the CAD modeler to maintain and satisfy these constraints throughout the optimization process.

Segmentation plays an important role during the reconstruction process to identify the boundaries of parts within an assembly point cloud. This process is manual in CATIA, where the user selects and identifies a set of points with the selection tools like the brush to segment a set of points. DesignX allows to automatically segment and identify the different sets of points to generate surfaces. In our proposed method, segmentation is incorporated within an optimization loop using a two-level filtering technique. Reverse engineering with CATIA system does not provide editable CAD models at the end of the RE process. The non-editable CAD part has no use when considering the late design modification. On the other hand, DesignX can retrieve an editable CAD model for certain features like revolution,

Criterion	New Reverse Engineering	CATIA	Design X
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
Registrations	✓	✓	✓
Late design changes	✓	-	-

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

extrusion, holes, etc. but this is not fully true when the part is complex with many features. This also requires some skills to fully retrieve an editable CAD model with DesignX, while our method already works on the features during the optimization process which makes it more suitable for the late design changes. Another strong point of our new RE is that the user can come back anytime to any step during the reconstruction process. The proposed RE does rely on the reference part as any reconstructed part used as a reference can further be fitted at the end of the optimization process. Users can also perform an overall global fitting for single or multiples parts at the end of the optimization if not well fitted previously. For instance, we can think of a scenario where the user fits different components of assembly with less accuracy first, and once the fitting is done, a global fitting can be performed at the end where all the components of the assembly will be fitted with higher accuracy. At this level, global fitting can make use of parameters that belongs to the 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.

Perspectives

The proposed approach is modular and each module can still be improved separately. In addition to the perspectives already mentioned in the different papers, here are additional ones. The current implementation is at the prototype level and it does not fully allow for real-time fitting. This is mainly due to the modularity of the implementation that calls external modules that need to exchange

with the optimization kernel. Thus, a better integration can drastically reduce the time spent to read/write/update the data in the various data structures of the different modules. In addition, in the current version, the thresholds used to tune the accuracy of the final results have been set up to quite low values so as to get very good fitting results. In the cases where optimization is performed with less accuracy, the obtained parameters can be rounded at the end of optimization. Parallelization can also be a good mean to speed up the overall process, while considering for instance that the parameters loops are executed step after step while letting the user free to interact and continue his/her work on the already fitted parts.

At the moment, 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 can further be 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. The fitting algorithm could also benefit from deep learning approaches to allow a faster fitting of parts. This could be interesting in the case of tracking the position of the robotic arm in Chapter 2 to update its digital twin.

Moreover, the created database (Chapter 3) with stored best configurations for the SA-based fitting is rather small and only contains few 3D CAD parts. However, the overall framework has been validated. For future perspectives, the database can be enlarged to also contain parametric models and their best configurations for the fitting of 2D sketches and assemblies. Sensitivity analysis performed at the start of the optimization can be integrated into the SA algorithm to track the sensitivities of parameters all along the optimization. Keeping track of the evolution of sensitivities, the system will automatically handle the less sensitive parameters.

Bibliography

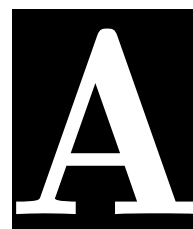
- [1] E. Bagci, “Reverse engineering applications for recovery of broken or worn parts and re-manufacturing: Three case studies,” *Advances in Engineering Software*, vol. 40, n^o. 6, p. 407–418, 2009.
- [2] A. Kumar, P. K. Jain et P. M. Pathak, “Reverse engineering in product manufacturing: an overview,” *DAAAM international scientific book*, vol. 39, p. 665–678, 2013.
- [3] “GrabCAD, 3D CAD model library,” <https://grabcad.com>, Accessed: 2021-05-15.
- [4] “TraceParts,” <https://www.traceparts.com/en>, Accessed: 2021-05-15.
- [5] “3DmodelSpace,” <https://www.3dmodelspace.com>, Accessed: 2021-05-15.

Note

A separate list of references has been placed within each chapter.

BIBLIOGRAPHY

APPENDIX



Annex

A.1 Developed prototype

A prototype of a plugin has been created for SolidWorks to provide a user interface for ease the application of the proposed RE methodology. . The plugin interface helps the users to choose optimization techniques and set different parameters taken from the GUI to perform the CAD-driven reverse engineering of a point cloud as shown in Figure 1. Multiples modules are integrated within the prototype plugin which calls the built-in functions e.g. MATLAB handles the SA algorithm, computes the distances and normals, and also filters the point cloud. SolidWorks is used to performs the successive updates of the geometries, to tessellate 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. All the modules are linked together to perform the fitting. The following sections detail the different steps for the usage of GUI to perform the CAD-driven reverse engineering of a point cloud.

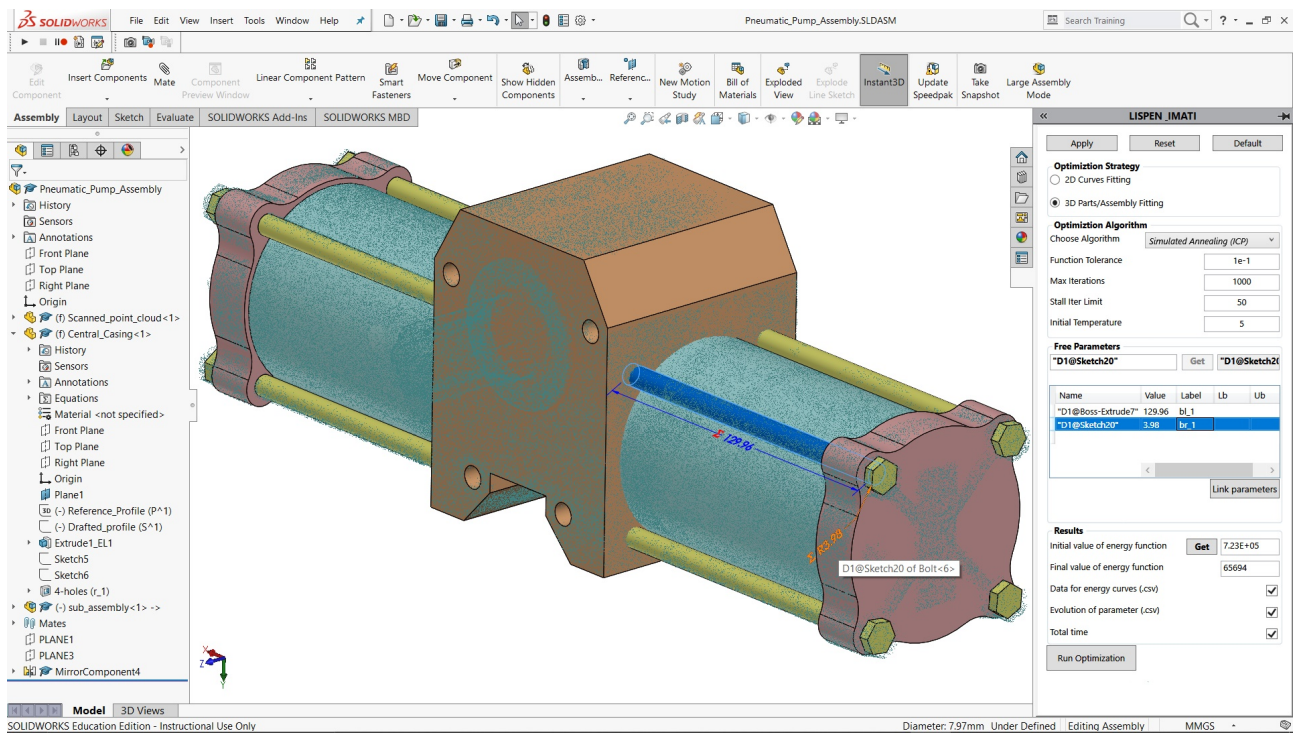


Figure 1 – User interface of prototype plugin for SolidWorks.

A.1.1 Optimization strategy

In the user interface, the first selection to be done is for the optimization strategy. Two options have been set for this selection criterion, one for the 2D curve fitting and the secondly fitting for the 3D parts/assembly as shown in Figure 2. Based on the choice, the optimization pipeline will choose its path either for the 2D profile fitting or 3D part fitting, and both are different from each other in terms of coordinates of the points. Profile fitting is based on 2D points, while 3D fitting is performed with 3D points.

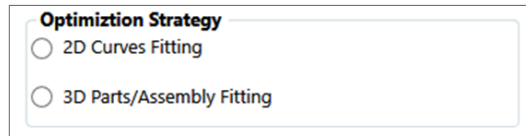


Figure 2 – Selection of optimization strategy.

A.1.2 Optimization algorithm

Once the optimization strategy is selected, the next step is to set the parameters of the optimization algorithms. As discussed previously, the proposed CAD-driven reverse engineering strategy is based on a Simulated Annealing algorithm, basic parameters shown in Figure 3 are required for the initialization. Apart from Simulated Annealing, there is also a possibility of *Fminseach* function for the minimization. For fewer parameters of CAD models, *Fminseach* function also shows promising results for fitting 2D and 3D parametric sketches and parts respectively.

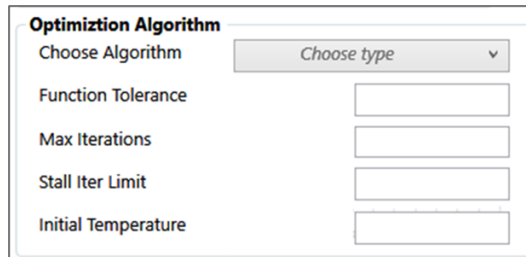


Figure 3 – Selection of optimization strategy.

As the proposed strategy can make use of ICP (iterative closest point) for the alignment and rotation, the selection of algorithm is provided with the option of ICP in the drop-down menu shown in Figure 4.

Depending on the requirement, the user can choose this optimization algorithm either with ICP or without it.

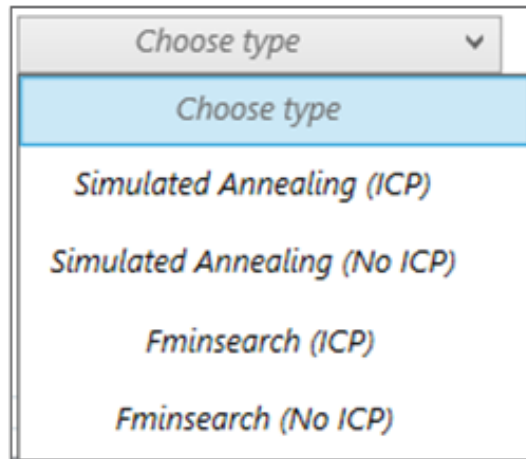


Figure 4 – Selection of optimization algorithm either with ICP or without it.

The values for *Function Tolerance*, *Max Iterations*, *Stall Iter Limit* and *Initial Temperature* can be adjusted and modified with the user input values or can be set with the default values using the default button in the user interface shown in Figure 5. The default values have been chosen based on series of experiments carried out for the reconstruction of multiple examples and are supposed to produce better results in most of the fitting cases.

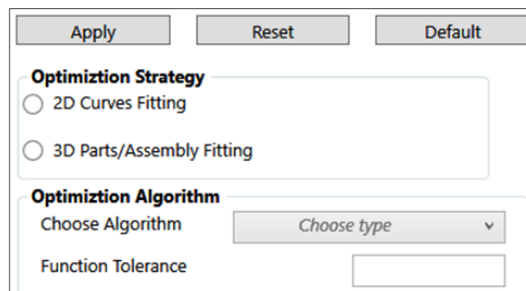


Figure 5 – Setting default values for the optimization.

A.1.3 Selection of free parameters

After setting the optimization algorithm, the user has to choose the free parameters from either *2D parametric sketch* or *3D part/assembly fitting*. These parameters will be optimized by the chosen algorithm throughout the optimization process until the 2D sketch or 3D part/assembly is not fully

fitted. The choice of selection of the free parameters still remains with the user. To select these parameters, the user needs to select the parameter’s dimension (highlighted in blue in Fig. 1) from the graphical interface of the CAD modeler and then click the “*Get*” bottom to retrieve its value and register this information in the list box as shown in Figure 6. Moreover, users can label different parameters to be optimized and can also provide lower (*Lb*) and upper (*Ub*) bounds for each parameter to force SA-based optimization to look for the solutions within given limits.

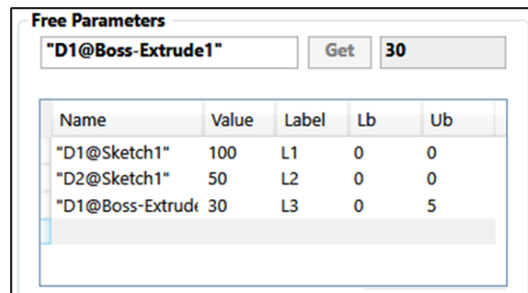


Figure 6 – Selecting free parameter’s values.

A.1.4 Results

Finally, to obtain the results, this section also requires some inputs from the user prior to executing optimization for the reconstruction process. After pre-arranging the CAD part or 2D sketch to the reference point cloud as explained in the previous chapters on the technical aspects of the fitting, the initial overall energy (Initial value of energy function) of the system can be retrieved by pressing the button “*Get*” as shown in Figure 7. Users can also save additional information on the energy curves,

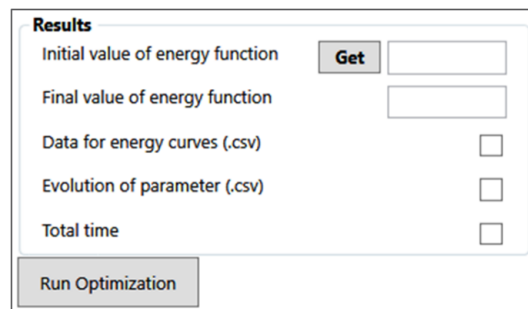


Figure 7 – Options selection for the results section.

the evolution of parameters and total time taken, by selecting different options. Once every section has been set, the user can now execute the optimization process by pressing the “*Run optimization*” button.

A.2 Application of the prototype plugin

For better clarify the usage of the plugin, the Sonotrode CAD part is reconstructed. This tutorial will help the user to better understand the functionalities and sequence of procedures to be followed. Following are the sequence of operations:

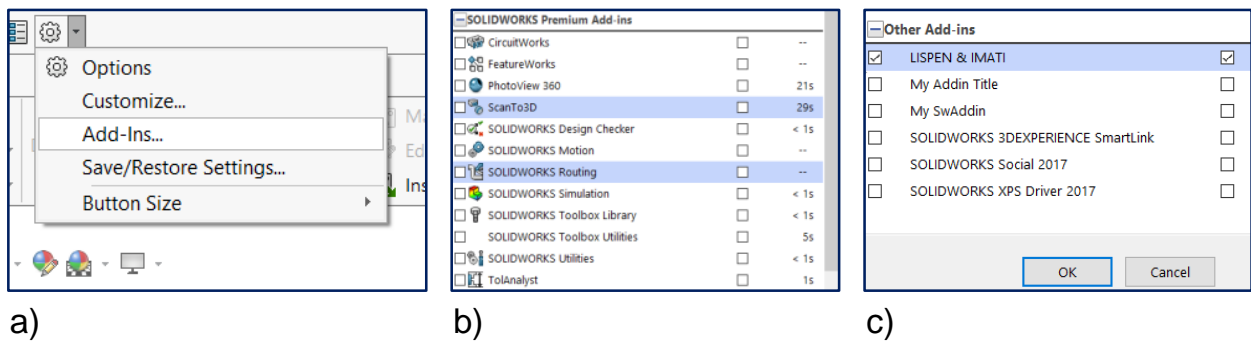


Figure 8 – Accessing routing plugin from SolidWorks.

A.2.1 Selection and prearrangement of the CAD model

Before starting the procedure, the add-in of SolidWorks, “*ScanTo3D*” must be activated from the options to visualize the point cloud in the SolidWorks. This can be accessed from the “*options*” (Fig. 8.a). In the “*Add-ins*” list “*ScanTo3D*” has to be selected as shown in Figure (Fig. 8.b). After the installation of the prototype plugin, it can be accessed from the “*Other Add-ins*” with the name “*LISPEN IMATI*” as shown in Figure (Fig. 8.c). Next, create an assembly containing the draft of the CAD part and the reference point cloud (fig. 8-a) by importing them into the assembly module of the SolidWorks. For this particular reconstruction procedure of Sonotrode, ICP is not used and the CAD part is well aligned in the center of the point cloud using the assembly tools (Fig. 9). Otherwise, if ICP is selected with the optimization algorithm, the user just needs to pre-arrange the CAD part with reference to the point cloud without good alignment. In this case, the pre-arrangement of the CAD part should be as close as possible with respect to both dimension and position to the given reference point cloud to avoid possible local minimum.

A.2. APPLICATION OF THE PROTOTYPE PLUGIN

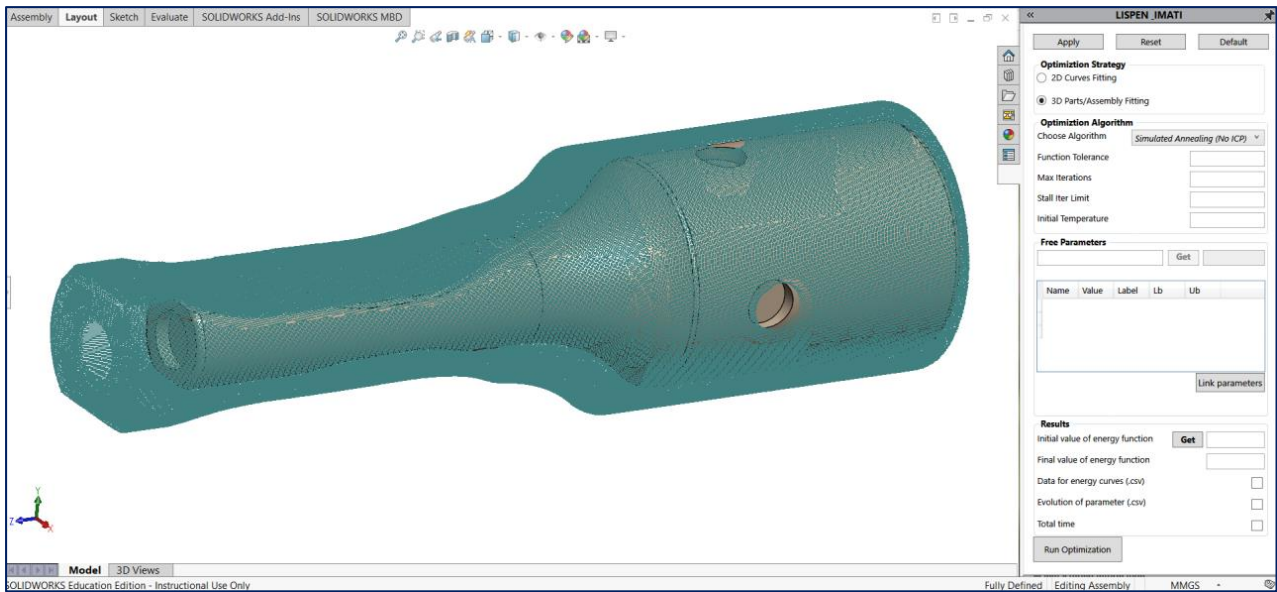


Figure 9 – Opening the CAD model and point cloud in the CAD modeler.

A.2.2 Setting optimizer (plugin values)

Once the CAD model and reference point cloud are opened and pre-arranged, the next step is to set the optimizer for the optimization of CAD model's parameters. Different steps involved are as under.

A.2.2.1 Optimization strategy

Depending on the requirements, the user can choose either “*2D curve fitting*” if he/she needs to perform fitting of 2D profile or “*3D parts/assembly fitting*” for the 3D parts/assemblies. In this example, 3D CAD part is to be fitted so, the second option i.e. “*3D parts/assembly fitting*” is selected (Fig. 9) from the “*Optimization Strategy*”.

In the next step, the user is required to select the Optimization Algorithm. Here we choose, Simulated Annealing without ICP and the corresponding values of different parameters e.g. function tolerance, Max iteration, stall-iterations and initial temperature are selected. The input values for these parameters can either be set manually by the user or default values can be set by pressing the default button in the optimizer window which will assign default values to different parameters (Fig. 10). The default values have been identified based on series of fittings performed for a variety of CAD models. Otherwise, the user can also take these values from the best-identified configuration from the database of CAD models

A.2. APPLICATION OF THE PROTOTYPE PLUGIN

using similarity assessment as detailed in the Chapter 4.

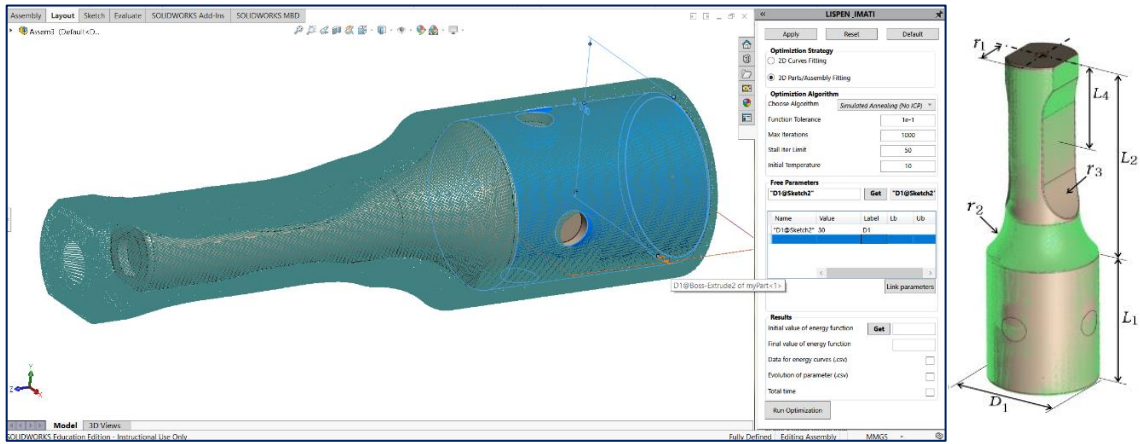


Figure 10 – Setting optimization algorithm with default parameters and selection of the free parameters from CAD model.

A.2.2.2 Free parameters

After setting the optimization algorithm, now the user is required to choose a set of free parameters of the CAD model from the GUI of SolidWorks to be optimized for the fitting. The iterative process will continue until the shape of the CAD part is fully fitted with the reference point cloud. To choose a parameter, the user has to select a dimension and click the “Get” button in the Free parameters section and then “Apply” button to get the parameters in the table list (Fig. 10). In the list, parameters are then labeled with new names. The lower and upper bounds are left open without any limits. The selection of free parameters for the Sonotrode has been detailed in section 5.4 of chapter 2 on the global fitting of a Sonotrode to a real scanned point cloud.

Once all the required parameters are selected, they are linked to text file of parameters to be used by the optimizer for writing and updating new parameters with every iteration of optimization. To link these parameters, the button “Link parameters” is clicked (Fig. 11). This will automatically generate a text file linking these parameters from SolidWorks to the Simulated Annealing algorithm.

A.2.2.3 Results

Finally, to get some results, the user can choose certain options like the initial value of energy that can be obtained by clicking the button “Get” next to it. The final value of the energy will automatically

A.2. APPLICATION OF THE PROTOTYPE PLUGIN

be computed as soon as the optimization stops. Users can decide to save the data for energy curves, total time and the data for the evolution of parameters in CSV format by selecting these options and then start the optimization by clicking "Run optimization". The fitted Sonotrode CAD part with the new dimensions is shown in Figure. 12.

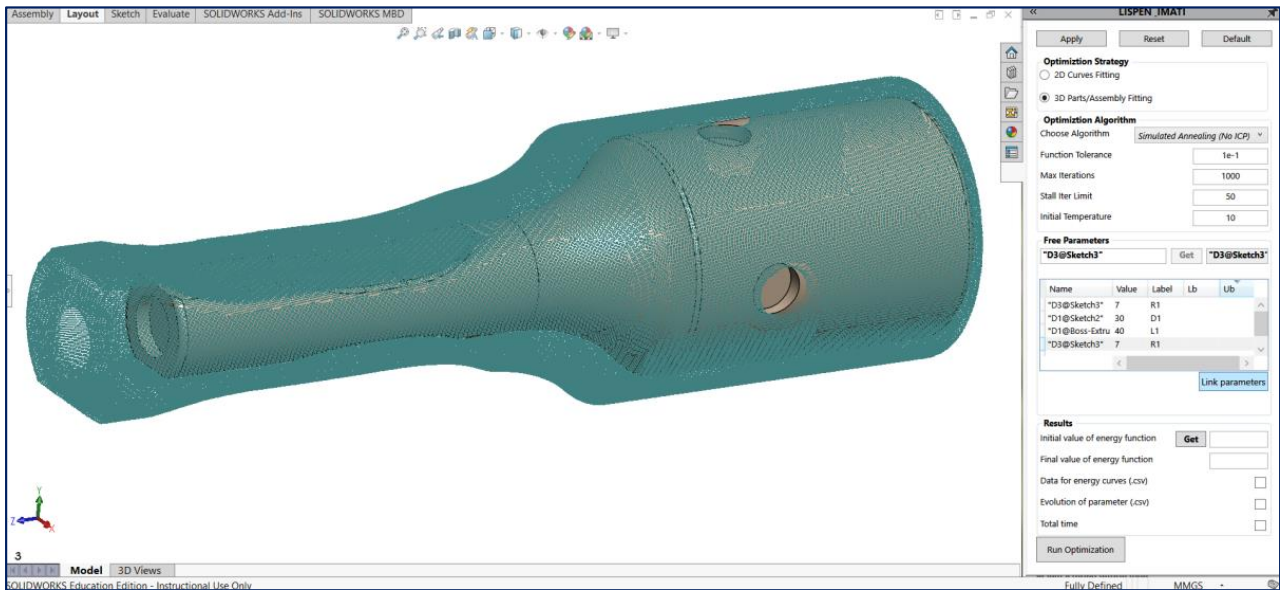


Figure 11 – Linking parameters of CAD model to Simulated Annealing to optimize during the optimization process.

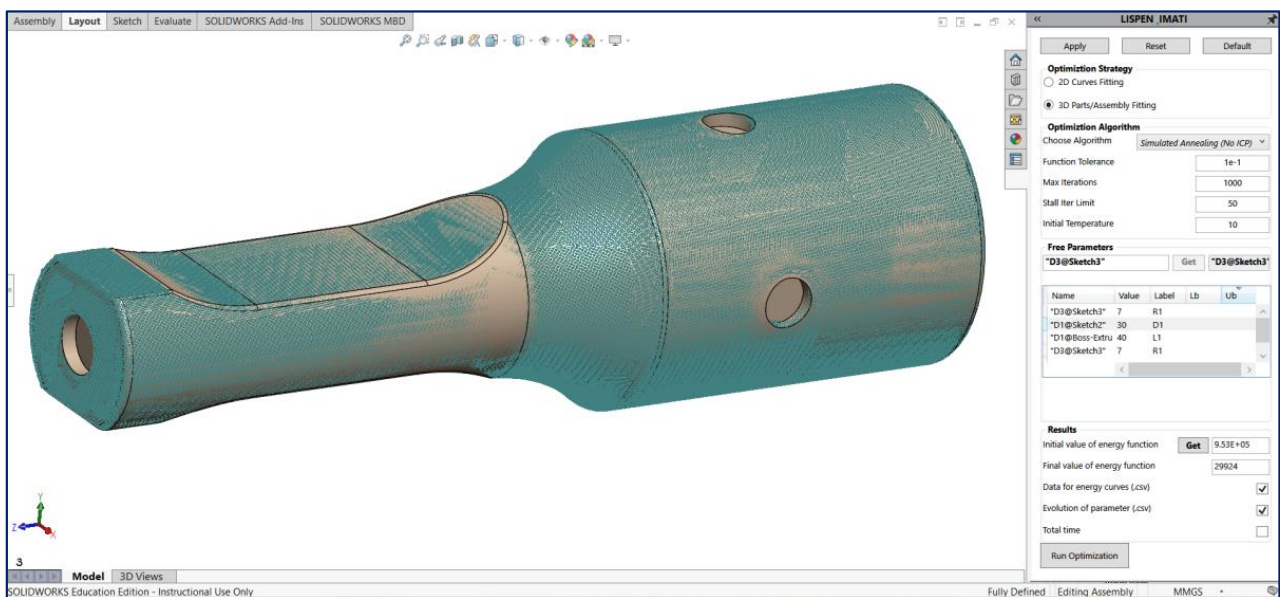


Figure 12 – Fitted Sonotrode after the optimization.

Sommario : Nonostante l'esistenza di molte tecniche di Reverse Engineering per ricostruire modelli digitali 3D di oggetti reali, pochissimi metodi sono in grado di ottenere in modo diretto ed efficiente la ricostruzione di modelli CAD di assiemi di parti meccaniche modificabili e quindi pienamente utilizzabili nelle varie fasi del Processo di Sviluppo di Prodotto (PDP). In assenza di strumenti di segmentazione adeguati, questi approcci hanno difficoltà ad identificare e ricostruire il modello delle diverse parti che compongono l'assieme. Questa tesi mira a definire una nuova tecnica di Reverse Engineering per la ricostruzione di modelli CAD parametrici mediante l'uso di modelli di riferimento. L'originalità risiede nell'uso di un processo di ottimizzazione che sfrutta la tecnica di simulated annealing ed un filtraggio a due livelli in grado di catturare e gestire i confini delle singole geometrie all'interno della nuvola di punti completa per consentire il rilevamento dell'interfaccia e l'adattamento locale del modello di una singola parte alla nuvola di punti. Il metodo proposto è applicabile a vari tipi di geometrie, ovvero sezioni 2D, parti singole e assemblati ed utilizza vari tipi di dati (nuvole di punti, modelli CAD, parametri relativi alle geometrie da ricostruire e al processo di adattamento).

Il metodo proposto presenta quindi un approccio modulare in due fasi. La prima fase mira ad identificare le interfacce tra le parti sfruttando e combinando un insieme di caratteristiche all'interno dei dati disponibili (nuvole di punti e modelli CAD). La seconda fase gestisce l'adattamento dei modelli CAD di riferimento alle corrispondenti porzioni delle nuvole di punti scansionate. Il metodo integra l'analisi di sensitività per caratterizzare l'impatto delle variazioni nei parametri del modello CAD di riferimento sull'evoluzione della deviazione tra il modello CAD stesso e la corrispondente nuvola di punti. Al fine di facilitarne l'utilizzo, la tesi propone inoltre un approccio case-based per l'identificazione ed utilizzo delle configurazioni ottimali dei parametri in base alla forma dell'oggetto considerato.

L'approccio proposto è stato valutato utilizzando sia nuvole di punti ottenute da reali scansioni che nuvole di punti generate virtualmente (as-scanned) che incorporano diversi artefatti così come potrebbero apparire nel caso di acquisizione con un vero scanner. I risultati sono di interesse per diversi scenari applicativi correlati all'Industria 4.0, che vanno dall'adattamento globale di una singola parte all'aggiornamento di un assieme completo con relativi vincoli tra le parti. Pertanto l'approccio proposto presenta caratteristiche idonee a supportare la coerenza tra un prodotto / sistema e il suo gemello digitale.

Parole chiave : reverse engineering, 2D and 3D fitting, Simulated Annealing, gemello digitale, industria 4.0, analisi di sensitività, modelli CAD parametrici, nuvole di punti, segmentazione.

Résumé : Même si de nombreuses techniques de rétro-ingénierie existent pour reconstruire des objets réels en 3D, très peu sont capables de traiter directement et efficacement la reconstruction de modèles CAO éditables d'assemblages de pièces mécaniques pouvant être utilisés dans les étapes du processus de développement de produits (PDP). En l'absence d'outils de segmentation adaptés, ces approches ont des difficultés à identifier dans le modèle reconstruit les différentes pièces qui composent l'assemblage. Cette thèse vise à développer une nouvelle technique de rétro-ingénierie pour la reconstruction de modèles CAO modifiables. L'originalité réside dans l'utilisation d'un processus de calage basé sur l'algorithme d'optimisation de recuit simulé et exploitant un filtrage à deux niveaux capable de capturer et de gérer les limites des géométries des pièces à l'intérieur du nuage de points global. Cette approche permet ainsi la détection des interfaces et l'ajustement local d'un modèle de pièce au nuage de points. La méthode proposée exploite différents types de données (par exemple, des nuages de points, des modèles CAO éventuellement stockés dans une base de données avec les meilleures configurations de paramètres associées pour le processus calage). L'approche est modulaire et intègre une analyse de sensibilité pour caractériser l'impact des variations des paramètres d'un modèle CAO sur l'évolution de la déviation entre le modèle CAO lui-même et le nuage de points à caler. L'évaluation de l'approche proposée est réalisée en utilisant à la fois des nuages de points scannés réels et des nuages de points générés virtuellement, qui comportent plusieurs artefacts pouvant apparaître avec un scanner réel. Les résultats couvrent plusieurs scénarios d'application liés à l'industrie 4.0, allant de l'ajustement global d'une seule pièce à la mise à jour d'une maquette numérique complète intégrant des contraintes d'assemblage. L'approche proposée est particulièrement adaptée pour l'aider au maintien de la cohérence entre un produit/système et son jumeau numérique.

Mots clés : rétro-ingénierie, calage 2D et 3D, recuit simulé, jumeau numérique, industrie 4.0, analyse de sensibilité, paramètres de modèles CAO, nuages de points scannés, segmentation.

Abstract : Even if many Reverse Engineering techniques exist to reconstruct real objects in 3D, very few are able to deal directly and efficiently with the reconstruction of editable CAD models of assemblies of mechanical parts that can be used in the stages of Product Development Processes (PDP). In the absence of suitable segmentation tools, these approaches struggle to identify and reconstruct model the different parts that make up the assembly. The thesis aims to develop a new Reverse Engineering technique for the reconstruction of editable CAD models of mechanical parts' assemblies. The originality lies in the use of a Simulated Annealing-based fitting technique optimization process that leverages a two-level filtering able to capture and manage the boundaries of the parts' geometries inside the overall point cloud to allow for interface detection and local fitting of a part template to the point cloud. The proposed method uses various types of data (e.g. clouds of points, CAD models possibly stored in database together with the associated best parameter configurations for the fitting process). The approach is modular and integrates a sensitivity analysis to characterize the impact of the variations of the parameters of a CAD model on the evolution of the deviation between the CAD model itself and the point cloud to be fitted. The evaluation of the proposed approach is performed using both real scanned point clouds and as-scanned virtually generated point clouds which incorporate several artifacts that could appear with a real scanner. Results cover several Industry 4.0 related application scenarios, ranging from the global fitting of a single part to the update of a complete Digital Mock-Up embedding assembly constraints. The proposed approach presents good capacities to help maintaining the coherence between a product/system and its digital twin.

Keywords : reverse engineering, 2D and 3D fitting, Simulated Annealing, digital twin, industry 4.0, sensitivity analysis, CAD model parameters, as-scanned point clouds, segmentation.