Title:
As-scanned Point clouds generation for Virtual Reverse Engineering of CAD Assembly Models

Authors:
Jérémy Montlahuc, jeremy.montlahuc@ensam.eu, Arts et Métiers, LISPEN, HeSam, France
Ghazanfar Ali Shah, ghazanfar_ali.shah@ensam.eu, Arts et Métiers, LISPEN, HeSam, France
Arnaud Polette, arnaud.polette@ensam.eu, Arts et Métiers, LISPEN, HeSam, France
Jean-Philippe Pernot, jean-philippe.pernot@ensam.eu, Arts et Métiers, LISPEN, HeSam, France

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

DOI: 10.14733/cadconfP.2018.117-122

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 [2][5]. 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 setting up 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 the acquisition and processing time can range from a few minutes to several hours. 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 acquisition 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 seconds.

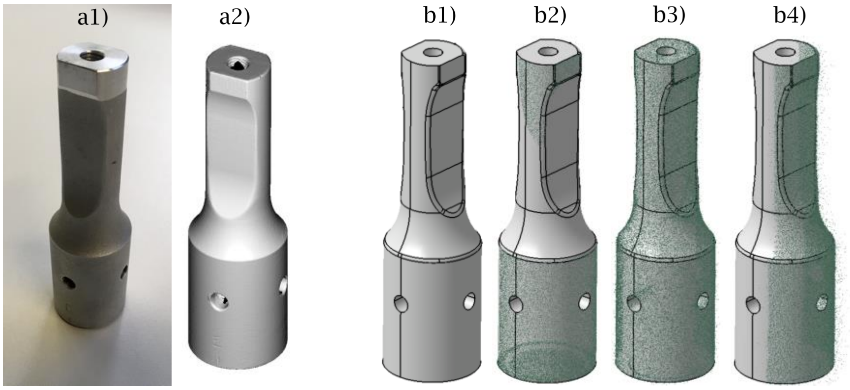

Fig. 1: Real sonotrode (a1) scanned to get point clouds that have been post-treated (a2). CAD model of a sonotrode (b1) virtually reverse engineered following three parameterizations of the proposed framework (b2-b4).

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 [1].

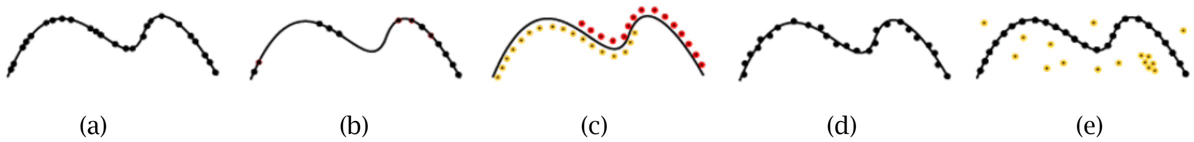

Fig. 2: Different artifacts: (a) non-uniform sampling, (b) missing data, (c) misaligned scans, (d) noisy data and (e) outliers [1].

This paper introduces a modular framework for the generation of as-scanned point clouds incorporating the above mentioned artifacts.

<u>Overall framework:</u>
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 not anymore watertight 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. The approach is modular and each module is controlled by parameters which are detailed in the next section.

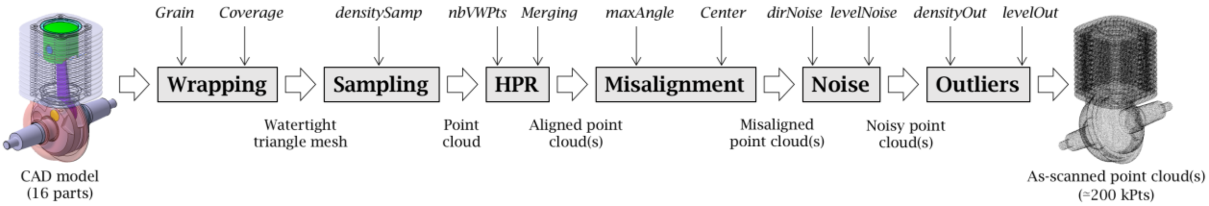

Fig. 3: Virtual Reverse Engineering modular framework generating as-scanned point clouds from CAD assembly models.

		Modules					
		Wrap-ping	Sampling	HPR	Misalignment	Noise	Outliers
Artifacts	Non-uniform sampling	Yes	Yes	Yes	-	-	-
	Missing data	Yes	Yes	Yes	-	-	-
	Misaligned scans	-	-	-	Yes	-	-
	Noisy data	-	-	-	-	Yes	-
	Outliers	-	-	-	-	-	Yes

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

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.

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 50μm. 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.

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 to perform the sampling.

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 Khalfaoui et al. [4], a set of *nbVWPts* viewpoints are selected from a predefined list of positions located on a sphere centered on the object to be virtually reverse engineered (Fig. 4.a). The removal of hidden points is performed using the approach of Katz et al. [3]. The adopted HPR operator determines the visible points of the point cloud, as viewed from a given viewpoint (Fig. 4.b and 4.c). 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.

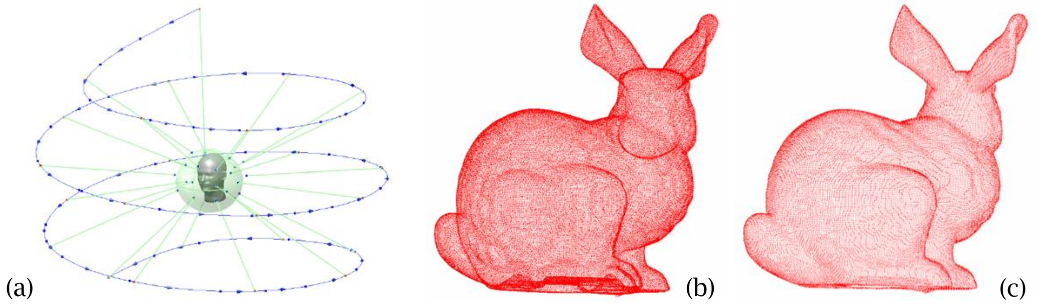

Fig. 4: Hidden Point Removal module: (a) Predefined acquisition viewpoints located on a sphere centered on the object to be virtually reverse-engineered [4]. (c) Visible points of an initial point cloud (b) as identified by the HPR operator running from a given viewpoint [3].

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 normal to the plan 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.

Noise

Being able to insert noise is an important feature of the proposed virtual Reverse Engineering framework [6]. 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. 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 × *levelNoise* × (rand – 0.5) where rand returns a single uniformly distributed random number in the interval [0, 1].

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 [1]. 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 × *levelOut* × (rand – 0.5).

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.

Tested configurations
To analyze the influence of the previously introduced control parameters and to show the potential of the proposed approach, three 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**
Wrapping	Grain	0.5mm	0.5mm	0.5mm
	wrapCoverage	0	0	0
Sampling	densitySamp	5	5	8
HPR	nbVWPts	1	4	2
	Merging	1	1	1
Misalignment	maxAngle	0°	0°	0°
	Center	Default	Default	Default
Noise	dirNoise	Normal	Normal	Normal
	levelNoise	0	0.03	0.1
Outliers	densityOut	0.1%	0.1%	0.1%
	levelOut	0	0.03	0.1

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

Evaluation criteria
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.

Following the configurations of Table 2, the virtual Reverse Engineering technique is applied on three different CAD models. Results are presented in Table 3 and in Figures 1, 5 and 6.

Models	**Parts (#)**	**Config.**	**Points (#)**	**Deviation (mm)**	**Coverage (%)**
Sonotrode	1	1	18 936	0.000 (0.0)	30.8
		2	61 327	0.033 (2.2)	76.9
		3	44 631	0.127 (7.5)	53.8
Compressor	16	1	54 831	0.00012 (1.35)	44.7
		2	91 950	0.00302 (2.36)	9.7
		3	163 241	0.07916 (8.48)	11.5
Engine	82	1	106 475	0.00023 (1.14)	11.7
		2	109 173	0.01145 (3.85)	14.2
		3	158 042	0.10563 (8.27)	16.0

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

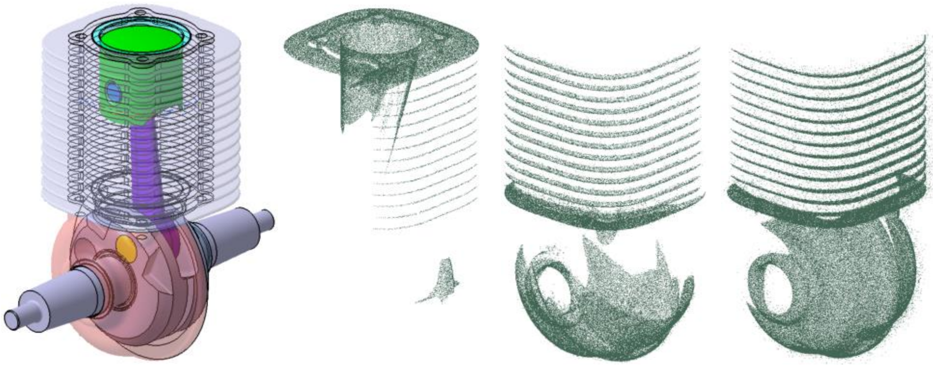

Fig. 5: CAD model of a compressor reversed engineered with configurations 1 to 3.

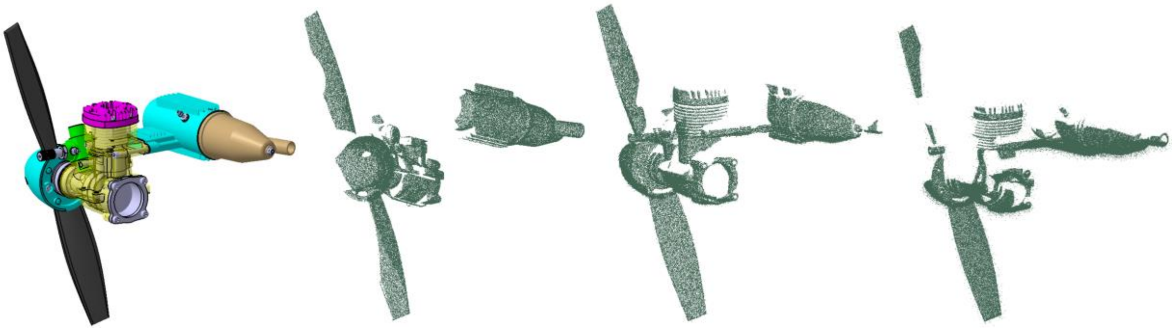

Fig. 6: CAD model of a RC engine reversed engineered with configurations 1 to 3.

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. 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 AI applications.

References:

[1] 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

[2] 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

[3] 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

[4] 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

[5] 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.

[6] 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.