Title:
Reverse Engineering for Aeronautics: Study on Parts Semantic Segmentation

Authors:
Philippe Williatte, philippe.williatte@utc.fr, Université de Technologie de Compiègne
Alexandre Durupt, alexandre.durupt@utc.fr, Université de Technologie de Compiègne, France.
Sébastien Remy, sebastien.remy@utt.fr, Université de Technologie de Troyes, France.
Matthieu Bricogne, matthieu.bricogne@utc.fr, Université de Technologie de Compiègne, France.

Keywords: Reverse-Engineering, CAO, Mesh, Segmentation, Machine-Learning

DOI: 10.14733/cadconfP.2022.170-178

Introduction:
Reverse Engineering (RE) is an activity which consists in digitizing a real part in order to create a numerical or virtual model of it [8]. It is conducted on components that do not have any Computer Aided Design (CAD) model, or only a semantically poor 3D representation, such as a mesh or a fixed resulting body. Main industrial applications for RE consist in CAD model re-design from digitized data for new product development or downward application such as Computer aided Manufacturing or simulation.

Aeronautical components present several challenges for RE activities, such as complex structures and shapes, large data volumes, and a high need of precision in CAD models rebuilt. Moreover, semantic segmentation (i.e., decomposition of a mesh into meaningful regions) of complex shapes that represent freeform surfaces with aerodynamic properties remains an area of research.

This paper presents a study on semantic segmentation for complex aeronautical Parts. Machine Learning and Deep-learning model-based segmentation methods are evaluated on a set of real aircraft engine parts.

State in the art in Reverse Engineering:

THE RE PROCESS
According to [5], most common RE frameworks can be decomposed in following steps : (a) Digitizing; (b) Pre-Processing; (c) Segmentation; (d) Modelling.

Digitizing a real object consists in creating a 3D virtual representation of it (see [2] for existing methods). The result usually takes the form of a 3D mesh or a point cloud (list of point coordinates). It should be noted that, for some RE needs, 3D data may come directly from tessellated CAD files and not from real world components.

Pre-Processing steps like decimation of the number of acquired points, mesh generation, surface smoothing, etc.,…, are operations available in most acquisition software.

Segmentation is the process of sub-dividing a 3D mesh into distinct regions called segments. Segmentation is called semantic when each identified region can be easily associated with a construction operation or a specific surface [5]. Complete surveys can be found in [10], [14] and [18].

Modelling operations are used to generate 3D models compatible with CAD-CAM applications. *Freeform* approaches are commonly distinguished from *feature-based* methods. The former is an explicit modelling approach, which consist in describing a solid by its surfaces, and usually result in a “frozen” or “dead” solid. The term *features* is widely and differently used in the literature for implicit modelling which seeks to recover a parametric and semantic model by using a procedural approach with sequences of constructions operations with parenting relations. Here, a feature will be considered as generic shape

with which designer can associate certain attributes and knowledge useful for reasoning about the product [17].

3D MESH SEGMENTATION

Semantic of models is all information in addition to its geometry. More precisely, local semantics are information on the individual parts which composed the model, as functional specifications, modeling methods, general Product Manufacturing Information (PMI), and design intents. In a RE process, the objective of the segmentation is to cluster points (vertices) with similar characteristics into homogeneous meaningful regions [18]. Segmentation is considered as semantic if computed segments can be associated with local features of the definition model.

Many methods are based on shape analysis that extract geometric intrinsic information on each vertices of the mesh, and cluster points with similar properties, and/or divide the mesh by identifying edges between regions. Most of the technics could not be considered as straightforward semantic segmentation methods, in the sense that no label are attributed to segment, which can hardly be associated with features. On the other hand, model-based methods use the analytic definition of quadric shapes to identify and group vertices as part of a primitive geometry. Methods usually detect the type of surface, and then compute the primitive generic (canonic) equation with a classic least square fitting algorithm [1], [4], [16]. Because 85% of engineering CAD models can be represented by associations of primitive shapes [6], methods for complex surfaces processing (as [3], [17]) have not been intensively studied in the literature, especially for mesh semantic segmentation. According to [19], non-regular shape identification in raw data would make shape reuse possibilities wider, but freeform features should not depend on a predefined library. Based on that assessment, machine/deep learning technologies are applied to RE purposes for several applications as global mesh labeling [7], [9], [21], primitives fitting improvement [12], or parametric curves and surfaces approximation [11]. As opposed to purely mathematical model fitting and geometric reasoning, deeper methods extract feature descriptors during a supervised learning phase to apply semantic segmentation and labeling [15]. Besides not being restricted to generic shapes, machine/deep learning techniques are usually presented as outperforming geometric reasoning in term of completeness, noise robustness, and generalization [20]. However, those are known to require high computation capacities and long processing (learning) time, when shape descriptors should meet certain requirements in terms of simplicity, easiness of calculation and processing speed [13].

The review of RE techniques reveals limitations for RE activities on aeronautic components, especially for the segmentation step. Local features with complex shapes (such as blades) can hardly be identified and isolated with most common mesh segmentation methods. Therefore, CAD surfaces/volumes modeling from aeronautical Parts digitized data is a complex and time-consuming process, even for skilled users. To overcome the complexity of RE activities for aeronautical components, next sections present a study on specific semantic segmentation methods and tools for complex shapes identification and semantic segmentation of Part meshes.

Studies on semantic segmentation for aeronautic Parts:

CONTEXT AND CHALLENGES

Following studies are part of researches carried out in the context of a PhD thesis in partnership with industrials from the aeronautic sector. In our applications, RE for aeronautical parts is characterized by heavy meshes (up to 10M vertices) that are to be re-modelled with high precision and details. In this work, we will only consider the segmentation step of the overall RE process, considering that digitized mesh quality and conformity is ensured by the metrology department. Moreover, modelling steps consists of b-rep surfaces fitting and/or deformable templates fitting to the mesh, which can be completed with sufficient results by commercial RE software [5]. Three methods for semantic mesh segmentation will be studied and compared on real aircrafts engines parts. First two are model-based

methods using Ordinary-Least-Squares nonlinear regression[1] and Gaussian Process regression[2] for complex surface models approximation. The third method consists in using the neural network Pointnet++ [15] on a private dataset to evaluate Deep-learning capacities for semantic segmentations in the aeronautic area.

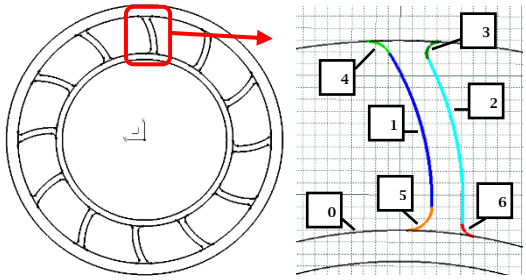

Fig. 1: Simplified sketch of an aircraft engine exhaust casing and its blade's topology.

KNOWLEDGE CAPITALIZATION ON 3D DATA

Component presented in Fig. 1 is taken as an example in the following:

$Part_{def}$ refers to a definition CAD model, and S_{def_i} ($i \in [1,n]$) to n local surfaces (ie. features to be identified in a mesh). $Part_{scan}$ is the digitized model (mesh) of one physical component. $Part_{mesh}$ and S_{mesh_i} refers to tessellated 3D models of $Part_{def}$ and S_{def_i}. Finally, $segm_i$ are sub-meshes of $Part_{scan}$. If the semantic segmentation performs well, $segm_i$ should be the image of S_{def_i} in $Part_{scan}$. Methods in this study are based on a capitalization (learning) phase that computes "features signatures", called $sign_i$, that consist of shape descriptors and feature's metadata.

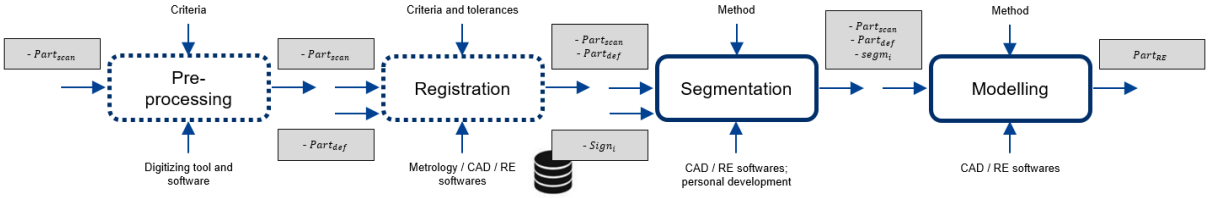

Fig. 2: overall RE process.

Polynomial implicit functions of surfaces

This method is a model-based segmentation technique that seeks to fit an algebraic surface into the point cloud. Just like the well-known RANSAC method (cf. [1], [16]) iteratively optimize primitives fitting to regions of points in a Cartesian space, our method try to fit more complex surfaces defined by high order polynomial implicit functions.

An implicit function f_d (with degree d) of a surface S_{def} is defined as follow:

$$S_{def} = \{p(x,y,z) \mid f_d(p) = 1\}, \quad f: \mathbb{R}^3 \to \mathbb{R}, \quad f_d(x,y,z) = \sum_{d_x=1}^{d} \sum_{d_y=1}^{d} \sum_{d_z=1}^{d} \alpha_{d_x,d_y,d_z} \, x^{d_x} y^{d_y} z^{d_z} \qquad (1)$$

$$\text{with } (d_x + d_y + d_z \leq d)$$

This mean, for any point p_j of $Part_{scan}$, and for an admitted threshold ε:

$$p_j \in segm_i \quad \text{if and only if} \quad \left| f_d(p_j) - 1 \right| < \varepsilon \qquad (2)$$

[1] https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb
[2] https://towardsdatascience.com/gaussian-process-regression-from-first-principles-833f4aa5f842

$f_{i,d}$ parameters are approximated with Ordinary Least Square polynomial regression[3] on S_{mesh_i} points (ie. vertex), and represent the shape descriptor (called OLS model) of $sign_i$. Model fitting to random data points consists of a rigid transformation optimization that minimize the squared sum of predictions errors. On a set of m points, parameters $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$ optimization is computed to minimize:

$$cost_i = \sum_{j=0}^{m}\left(f_{i,d}(x_{t,j}, y_{t,j}, z_{t,j}) - 1\right)^2 \quad , \quad \begin{pmatrix} x_t \\ y_t \\ z_t \\ 1 \end{pmatrix} = M.p = \begin{pmatrix} \theta_{xx} & \theta_{xy} & \theta_{xz} & t_x \\ \theta_{yx} & \theta_{yy} & \theta_{yz} & t_y \\ \theta_{zx} & \theta_{zy} & \theta_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3)$$

M is the transformation matrix computed with parameters $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$.

Whether the minimized $cost_i$ with optimized parameters is under an admitted threshold or not, we can determined if randomly selected data points can be labeled as S_{def_i}. By iteratively trying to fit candidate models $f_{i,d}$ to input data points, best label is attributed and the semantic segmentation of the mesh is performed by growing segments to neighboring points (with respect to equation (2)).

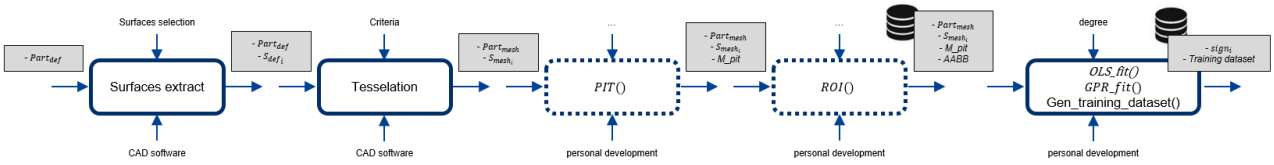

Fig. 3: method for feature signature.

PIT() refers to the function that applies the Principal Inertia Transform to S_{mesh} (which is used for stable pose for model approximation, and initial pose for model fitting). ROI() computes the Axis Aligned Bounding Box (AABB), which gives general information on features dimensions.

The methodology for the study is as follow:

(a) Signatures capitalization on several features (S_{def_i}) of several Parts. (cf. Fig. 3)

(b) Model analysis and evaluation: predictions results when approximating $f_{i,d}$ on tessellated features are analyzed for $4 < d < 8$ (equation (1)). For the best degrees of fitting, we compute $segm_i$ on $Part_{mesh}$ (equation (2)). The comparison of $segm_i$ and S_{mesh_i} show the segregations capability of the OLS surface model.

(c) Method evaluation: A random rigid transformation is applied to S_{mesh_i}. Competing $f_{i,d}$ are fitted on randomly selected local regions (equation (3)). The capacity of the OLS model to label a feature and extract corresponding segment from the global mesh is determined by the percentage of labeling success on a high number of try (if a local regions is correctly labelled, predictions on neighboring points is used to grow the segment.)

(d) The same operation is done with local regions of *scan* (for now manually selected).

Gaussian Process Regression

The second method doesn't use a specific model like (1) to approximate a function f_d on data points, but rather uses Gaussian Process Regression (GPR). Gaussian process is a stochastic supervised learning tool that defines a distribution over a function. Advantages in using GPR instead of Implicit functions is that there are less parametric and no parameters choices like degree are needed.

The idea behind this method is quite similar to the first one: by approximating $gpr(x, y, z) = 1$ (called GPR model) over S_{mesh_i}, a fitting cost (3) can be computed for local regions labelling. Applied

[3] https://www.statsmodels.org/stable/generated/statsmodels.formula.api.ols.html

methodology for evaluation is very similar to the previous one, except Gaussian Process Regression[4] is used instead of OLS polynomial approximation.

Deep Learning

Nowadays, researches have shown the potential of deep-learning in 3D geometry processing [20]. Tools as Artificial Neural Network (ANN) are able to learn deep shapes descriptors and apply semantic segmentation without shape typology-related limitations. Pointnet ++ ANN [15] is used for simple point-clouds segmentation by recognizing local features and adding labels to each point. However, to our knowledge, very few works show applications for real industrial needs. For comparative study with previous methods, we tried to use Pointnet++[5] for the semantic segmentation of our example Part.

First, we reproduced results of the research paper [15] on the public dataset Shapenet[6]. Then, we developed a program for private dataset generation. The dataset consist of $Part_{mesh}$ point clouds with associated labels to each vertex (label 1 if the vertex is part of any S_{mesh_1}, same for any capitalized features. Otherwise, vertex label is set to 0). Thousands of training files are generated by applying random rigid transform to data points, adding Gaussian noise to coordinates, and making random permutation between lines. After training and testing Pointnet++ model on the dataset, we achieved promising results.

In the next section, results of the study are presented, and methods are discussed.

Results and discussions:

First, we analyzed polynomial and Gaussian Process Regressions models approximations on S_{mesh_i} with Mean Squared Error, Standard Deviation and Range of predictions errors:

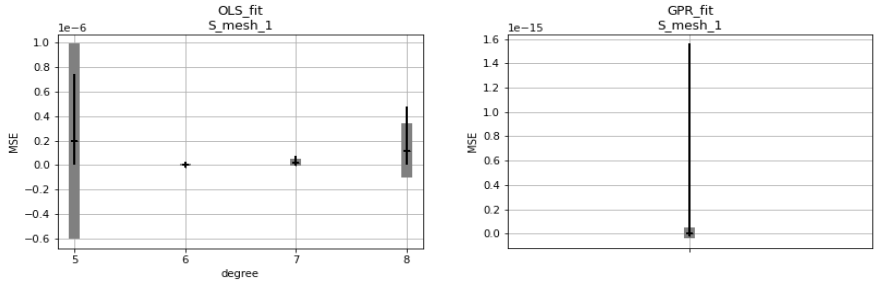

Fig. 4: model approximation analysis.

Computation time on ~4000 points is approximately 4 seconds with OLS model, and 30 seconds for GPR models. Predictions time are respectively 0.02 and 30 seconds

OLS and GPR models can approximate every different local surfaces of our studies with high precisions. Because both models are intrinsically unbounded, false positives can be solutions of equation (2) when results are predicted on every vertices of a global mesh, and be added to $segm_i$. $Part_{mesh}$ segmentations results (Fig. 5) show the model's segregation capacities: segmentations results with OLS models have high rates of false positive. For example, those represent up to 20% for $segm_4$ and $segm_5$. On the other hand, GPR models have very low false positive rates, underlying their high segregation capabilities. However, Gaussian Processes lose efficiency in high dimensional spaces. This results in a much higher computation time for mesh segmentation with GPR models than with OLS models (for example, computing $segm_1$ on $Part_{mesh}$ (equation (2)) takes less than 1 second with OLS model, and approximately 2800 seconds with GPR model.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html
[5] https://github.com/yanx27/Pointnet_Pointnet2_pytorch
[6] https://shapenet.cs.stanford.edu/media/shapenetcore_partanno_segmentation_benchmark_

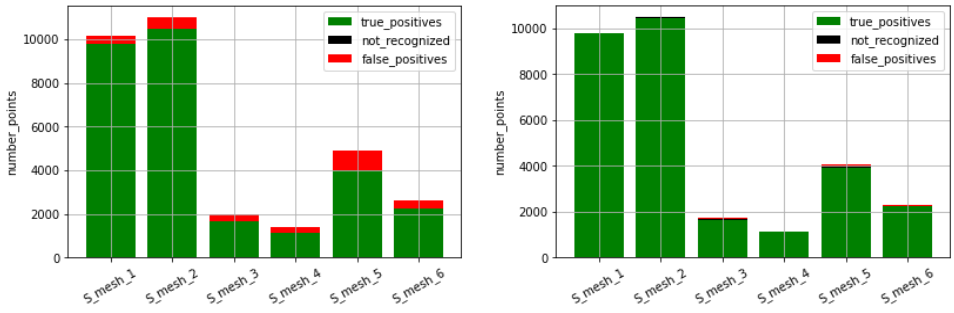

Fig. 5: Models evaluations.

Next stage consists in evaluating if approximated models can precisely be fitted on features which have been randomly transformed. Result for 100 try on each feature are presented in Table 1.

Feature label	Model	Result (success / try)	Average optimization time
1	OLS	90%	12
	GPR	80%	448
2	OLS	95%	13
	GPR	85%	757
3	OLS	70%	4
	GPR	100%	265

Tab. 1: model fitting on transformed features results.

OLS model fitting time makes it the most suitable model for the method. In the last experiment, random local regions (inside a sphere of radius 5 to 100 mm) of the global mesh are extracted and each capitalized models are fitted on it. Labeling can be considered as a success if the best fitting score correspond to the actual feature's OLS model, and the segment is correctly extended to neighboring

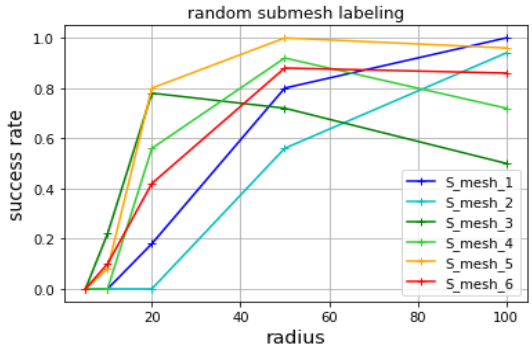

Fig. 6: local regions labeling and segment generation with OLS model fitting method.

points.

OLS model fitting method proved to be valid for a semantic segmentation of aeronautical Part meshes, but still requires improvement for more reliability. Some features are wrongly labelled if the originally selected region is too small, or features with similar shapes can be confused. However, the method

mostly succeeds to identify, recognize, and compute segments of local complex features in heavy meshes.

The last studied method for semantic segmentation of aeronautical meshes is the use of the ANN Pointnet++. Used dataset is generated from $Part_{mesh}$ and is composed of 3000 point clouds with each around 900 000 vertices points. Each blade is decomposed in 4 features, remaining vertices of the model are set to label 0. Training took approximately 36 hours for six epochs, and was stopped because evaluations metrics were not evolving. Figure 7 illustrates the results on a test data: For labels 1 to 4, true positives rates are respectively 5.8%, 6.03%, 3.47% and 0%. 83.4% of point which have been labelled 0 (brown points) are true positives. Segmentations results of the blades features is still poor, but most false positive are located in the neighborhood of the actual features.

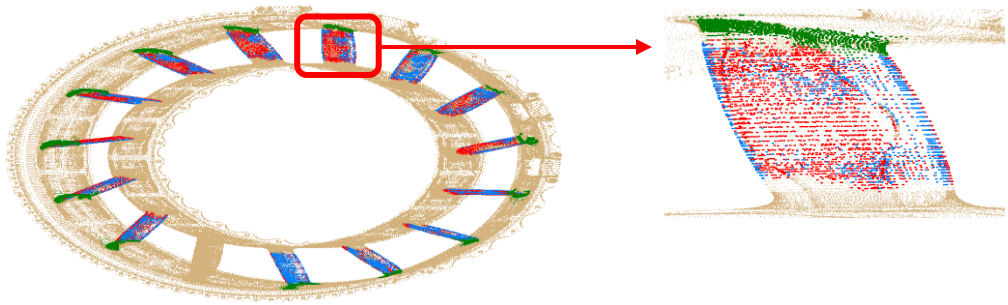

Fig. 7: Part segmentation results with Pointnet++.

Several limitations about our experimentation method should be considered: first, the limited number of epochs; then the use of Pointnet++ witout any parameter changes (as MLP tuning, or labels weights); and finally the fact that training data are highly unbalanced (label 0 represents approximately 83% of data points). However, this first experiment of using the ANN Pointnet++ for aeronautical components meshes segmentation shows promising results, as it can detect the presence of specific features with complex shapes in the right area. Moreover, as previous method, deep shape descriptors are robust to Euclidean transformations and works well for multi-instantiated features. Further experimentations and Neural Network parametrization will hopefully give better results. Nevertheless, the use of such technology for industrial needs is questionable, as it requires great computations capacities and time.

Conclusions:

We presented a study on semantic segmentation for Reverse Engineering activities on aeronautical components. We compared different technologies for subdividing 3D meshes into semantically meaningful sub-meshes that represent features of the definition CAD model. Our major proposition is the use of surface models using high degree analytic functions approximation and Gaussian process regression to extract and recover geometric information on complex shapes. By capitalizing surfaces models on definitions CAD 3D models, those can later be used as shape descriptors for sub-mesh labeling and global mesh segmentations, allowing semi-automated processing of complex aeronautic surfaces in heavy digitized data. However, this method still needs extensive testing and improvements to avoid errors when used to identify small features, or features with very similar shapes.

Although incomplete, we introduced a study on Deep learning applications in industrial context by testing the state-of-the-art Artificial Neural Network Pointnet++ for point clouds segmentation on a private dataset. Results are very promising, but actual use of such technology in industrial context would require optimizations for faster computation.

Further improvements of studied methods and tools will hopefully result in the development of performant shape descriptors able to identify complex features in Part meshes. The semi-automated semantic segmentation of digitized data will greatly simplify RE activities for aeronautic components.

References

[1] Attene, M.; Falcidieno, B.; Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives, Visual Computer, 22(3), 2006, 181–193. https://doi.org/10.1007/s00371-006-0375-x

[2] Barbero, B. R.; Ureta, E. S.: Comparative study of different digitization techniques and their accuracy, CAD Computer Aided Design, 43(2), 2011, 188–206. https://doi.org/10.1016/j.cad.2010.11.005

[3] Ben Makhlouf, A.; Louhichi, B.; Mahjoub, M. A.; Deneux, D.: Reconstruction of a CAD model from the deformed mesh using B-spline surfaces. International Journal of Computer Integrated Manufacturing, 32(7), 669–681. https://doi.org/10.1080/0951192X.2019.1599442

[4] Bénière, R.; Subsol, G.; Gesquière, G.; le Breton, F.; Puech, W.: A comprehensive process of reverse engineering from 3D meshes to CAD models, Computer-Aided Design, 45(11), 2013, 1382–1393. https://doi.org/10.1016/j.cad.2013.06.004

[5] Bruneau, M.; Durupt, A.; Roucoules, L.; Pernot, J.-P.; Benoît, E.: Towards new processes to reverse engineering digital mock-ups from a set of heterogeneous data, Proceedings of the Ingegraph-ADM- AIP Primeca Conference, 201.

[6] Buonamici, F.; Carfagni, M.; Furferi, R.; Volpe, Y.; Governi, L.: Reverse engineering by CAD template fitting: study of a fast and robust template-fitting strategy, Engineering with Computers, 2020. https://doi.org/10.1007/s00366-020-00966-4

[7] Buonamici, F.; Carfagni, M.; Furferi, R.; Governi, L.; Lapini, A.; Volpe, Y.: Reverse engineering modeling methods and tools: a survey, Computer-Aided Design and Applications, 15(3), 2018, 443–464. https://doi.org/10.1080/16864360.2017.1397894

[8] Chivate, P. N.; Jablokow, A. G.: Solid-model generation from measured point data, Computer-Aided Design, 25(9), 1993, 587–600. https://doi.org/10.1016/0010-4485(93)90074-X

[9] Dekhtiar, J.; Durupt, A.; Bricogne, M.; Eynard, B.; Rowson, H.; Kiritsis, D.: Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study, Computers in Industry, 100, 2018, 227–243. https://doi.org/10.1016/j.compind.2018.04.005

[10] Dimitrov, A.; Gu, R.; Golparvar-Fard, M.: Non-Uniform B-Spline Surface Fitting from Unordered 3D Point Clouds for As-Built Modeling, Computer-Aided Civil and Infrastructure Engineering, 31(7), 2016, 483–498. https://doi.org/10.1111/mice.12192

[11] Durupt, A.; Bricogne, M.; Remy, S.; Troussier, N.; Rowson, H.; Belkadi, F.: An extended framework for knowledge modelling and reuse in reverse engineering projects, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 233(5), 2019, 1377–1389. https://doi.org/10.1177/0954405418789973

[12] Guo, K.; Zou, D.; Chen, X.: 3D Mesh Labeling via Deep Convolutional Neural Networks, ACM Transactions on Graphics, 35(1), 2015, 1-12. https://doi.org/10.1145/2835487

[13] Kaiser, A.; Ybanez Zepeda, J. A.; Boubekeur, T.: A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data, Computer Graphics Forum, 38(1), 2019, 167–196. https://doi.org/10.1111/cgf.13451

[14] Ke, Y.; Fan, S.; Zhu, W.; Li, A.; Liu, F.; Shi, X.: Feature-based reverse modeling strategies, Computer-Aided Design, 38(5), 2006, 485–506. https://doi.org/10.1016/j.cad.2005.12.002

[15] Laube, P.: Machine Learning Methods for Reverse Engineering of Defective Structured Surfaces, 2020. http://www.springer.com/series/16265

[16] Li, L.; Sung, M.; Dubrovina, A.; Yi, L., Guibas, L.: Supervised Fitting of Geometric Primitives to 3D Point Clouds, ACM SIGGRAPH 2011 Papers on - SIGGRAPH '11, 2019. https://doi.org/10.1145/1964921.1964947

[17] Mousa, M. H.: Matching 3D objects using principle curvatures descriptors, Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2011, 447–452. https://doi.org/10.1109/PACRIM.2011.6032935

[18] Nguyen, A.; Le, B.: 3D point cloud segmentation: A survey, 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2013, 225–230. https://doi.org/10.1109/RAM.2013.6758588

[19] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space, Advances in Neural Information Processing Systems, 2017, 5100-5109.

[20] Schnabel, R.; Wahl, R.; Klein, R.: Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum, 26(2), 2007, 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x

[21] Song, Y.; Vergeest, J. S. M.; Bronsvoort, W. F.: Fitting and Manipulating Freeform Shapes Using Templates, Journal of Computing and Information Science in Engineering, 5(2), 2005, 86–94. https://doi.org/10.1115/1.1875592

[22] Theologou, P.; Pratikakis, I.; Theoharis, T.: A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation, Computer Vision and Image Understanding, 135, 2015, 49–82. https://doi.org/10.1016/j.cviu.2014.12.008

[23] Vergeest, J. S. M.; Spanjaard, S.; Horva´th, I.; Jelier, J. J. O.: Fitting Freeform Shape Patterns to Scanned 3D Objects, Journal of Computing and Information Science in Engineering, 1(3), 2001, 218–224. https://doi.org/10.1115/1.1419197

[24] Xiao, Y.-P., Lai, Y.-K., Zhang, F.-L., Li, C., & Gao, L. (2020). A survey on deep geometry learning: From a representation perspective. Computational Visual Media, 6(2), 113-133. https://doi.org/10.1007/s41095-020-0174-8

[25] Zhiron, W.; Song, S.; Khosla, A.; Fisher Y.; Linguang, Z.; Xiaoo, T.; Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 1912-1920. https://doi.org/10.1109/CVPR.2015.7298801