EXPLICIT ANNOTATED 3D-CNN DEEP LEARNING OF GEOMETRIC PRIMITIVES INSTANCES

Hilbig, Arthur;
Holtzhausen, Stefan;
Paetzold-Byhain, Kristin

Technische Universität Dresden, Chair of Virtual Product Development

ABSTRACT

In reengineering technical components, the robust automation of reverse engineering (RE) could overcome the need for human supervision in the surface reconstruction process. Therefore, an enhanced computer-based geometric reasoning to derive tolerable surface deviations for reconstructing optimal surface models would promote a deeper geometric understanding of RE downstream processes. This approach integrates advanced surface information into a deep learning-based recognition framework by explicitly labeling geometric outliers and subsurface boundaries. For this purpose, a synthetic dataset is created that morphs nominal surface models to resemble the macroscopic surface pattern of physical components. For the detection of regular geometry primitives, a 3D-CNN is used to analyze the voxelized components based on signed distance field data. This explicit labeling approach enables surface fitting to derive suitable shape features that fulfill the underlying surface constraints.

Keywords: Reverse Engineering, Surface Reconstruction, Machine learning, Computer Aided Design (CAD), Artificial intelligence

Contact:
Hilbig, Arthur
Technische Universität Dresden
Germany
arthur.hilbig@tu-dresden.de

Cite this article: Hilbig, A., Holtzhausen, S., Paetzold-Byhain, K. (2023) 'Explicit Annotated 3D-CNN Deep Learning of Geometric Primitives Instances', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.178

1 INTRODUCTION

No common solution has yet emerged to automatize the reverse engineering tasks of technical components fully. The current process flow in reverse engineering and parametric surface reconstruction still requires cooperative methods with an expert and several additional algorithmic approximation steps due to a need for interactive human-machine interactions (Buonamici *et al.*, 2018).
Therefore, an automatic and contextual surface segmentation would be beneficial to adapt to the variations of surface finishes of real-world objects. The improved recognition rates of artificial neural networks in computer vision tasks satisfy these needs with convolutional architectures in a deep learning (DL) manner (Bici *et al.*, 2020). An enhanced computer-based geometric recognition and detection in RE with additional information for subsequent algorithmic processes would promote a more coherent assessment of the implicit parametrization by the technical components RE expert (Shah *et al.*, 2022).
Although many geometric detection approaches apply DL to the process step of surface reconstruction, other methods that allow a deeper geometric model understanding in the sense of inspection techniques for surface defects are yet to be known (Geng *et al.*, 2022). Mostly, DL approaches derive the geometric primitives of technical components without providing further insights regarding the surface quality and the estimation of surface deviations from the nominal geometry. Different conclusions on macroscopic surface patterns for the downstream sub-steps of surface reconstruction are thereby limited, which could enable more exact approximations of the model parametrization in the future. In addition, adjusting the fitting algorithms to irregular and tolerable dimensional deviations and statistical evaluations would need a better classification of which surface components can be considered ideal and which should be neglected for fitting.
Therefore, this research aims to combine advanced surface information about surface deviations into a DL recognition process in surface reconstruction that allows computer-based geometry parameterization to be less reliant on expert supervision and adaptive interventions in the algorithmic processes.

2 BACKGROUND AND RELATED WORK

2.1 Reverse engineering overview

In general, reverse engineering is employed in various application areas to derive and identify components and interactions of a system by transferring the system representation to a higher level of abstraction (Chikofsky and Cross II, 1990). From the technical product design and modeling perspective, it reverses the forward development process from ideation, requirements definition, implementation, and manufacturing, etc., to extract design artifacts and, more specifically, transform real-world technical components into technical models and concepts (Várady *et al.*, 1997). Mostly, the reconstruction into a parametrically usable geometry model is the ultimate objective, which is then reintroduced into a product development process and, above all, can be used in computer-aided design (CAD) (Várady *et al.*, 1997). The baseline process steps can be distinguished into data acquisition (digitalization, registration, postprocessing) and surface reconstruction, which consists of the major sub-steps of segmentation/classification, surface fitting, constrained fitting, and surface model creation (Schöne, 2009).

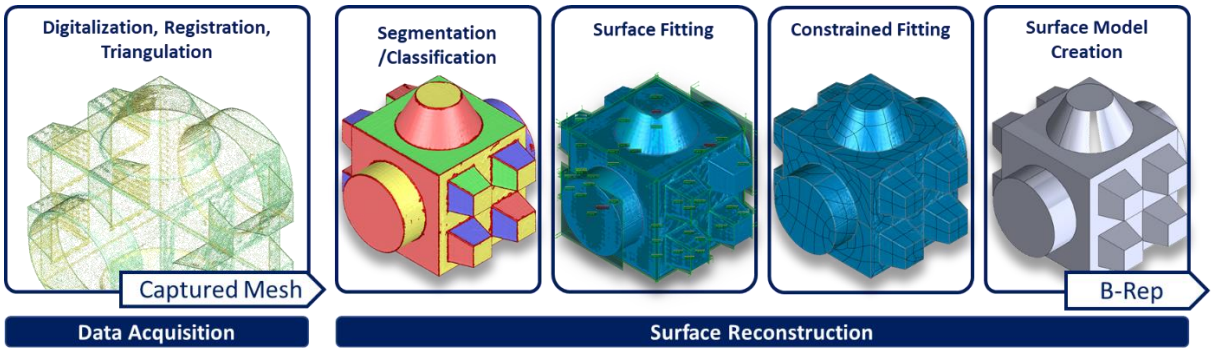

Figure 1. Processing pipeline of reverse engineering based on (Schöne, 2009)

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

The current state of research on RE automation is incomplete in deriving usable and consistent feature-based CAD models due to data and knowledge limitations for complex acquisition data in the area of robust surface reconstruction and geometric feature detection methods (Kaiser *et al.*, 2019). In particular, the real-world data fuzziness of manufacturing - and acquisition-related surface artifacts and deviations, as well as edge features, hinders generally valid, precise, and significant conclusions in the geometry analysis steps (Xie *et al.*, 2020). Thus, different sub-surfaces cannot comprehensively distinguish from other surface constituents in computer-based processing. Furthermore, on the one hand, static algorithm parameters require manual expert intervention in standard approaches, which makes generalizable parameter optimization impossible. On the other hand, current methods usually only consider a locally exact geometry representation or indirectly require global unambiguousness of the used geometry data to create valid CAD models. For this reason, the issue of surface reconstruction still needs to be conclusively resolved.

2.2 Common approaches for surface reconstruction

In the reverse engineering of technical components consisting of ruled geometric primitives (plane, cylinder, conus, sphere, torus), especially the deduction of the subsurfaces of technical components is error-prone since a variable adaptation to tessellation artifacts of the physical component data (noise, outliers, non-uniform sampling, misregistration, missing data) is difficult. Thereby, most approaches are implemented considering restrictions regarding the cardinality or number of segments, the resolution, and the discretization (Agathos *et al.*, 2007), which will be discussed in more detail in the following.

In practice, two main approaches (segmentation and direct estimation) have been established, which either observe local deviations of surface curvatures or directly estimate parametric partial surfaces by combining segmentation and surface generation. Other approaches include Hough Transforms (Woodford *et al.*, 2014) or Clustering algorithms (Attene *et al.*, 2006), which rely on prior knowledge like parameter limits or seed point selection. Recently, a variable output of a different set of segments using cluster analysis was investigated (Yan *et al.*, 2021), in which learned point-wise parametric properties are assigned to similar groups.

Methods that use surface curvatures for segment separation decompose the total surface into smooth sub-regions of equal geometric continuity. These classical region-based or edge-based segmentation methods fail due to the inhomogeneities of adjacent triangles because curvature measures as a transition condition always require an estimation of when a subregion is considered curvature continuous. Methods for direct parametric surface estimation and parameter approximation of a surface segment are primarily based on the Random Sample Consensus (RANSAC) algorithm, which allows outliers to be robustly identified and eliminated so that least squares methods determine the optimal segment parameters. For direct surface generation, EfficentRANSAC (Schnabel *et al.*, 2007) has established itself as a quasi-standard that allows decomposition into all major ruled primitives. A disadvantage is the random sampling and the threshold values for evaluating the model parameters, which can lead to sub-segmentation or a high number of artifacts (Fayolle and Pasko, 2016). Many novel knowledge-based RE approaches are based on the EfficentRANSAC, aiming for functional aspects (Qie *et al.*, 2021) and a production-oriented modification of technical component models (Ali *et al.*, 2013).

However, the quality of reconstruction results depends on an expert's careful and labor-intensive parameter optimization to overcome the algorithmic sensitivity to noise and outliers of the acquisition data. This leads, in practice, when using software tools like DesignX or Polyworks, to a sidestream mode of operation in RE where the automatization toolset is mostly neglected (Buonamici *et al.*, 2018).

Also, approaches that integrate geometric tolerancing (Kaisarlis *et al.*, 2008) have yet to be discovered in terms of automation of surface reconstruction. Though, surface deviation methods are applied in part inspection to compare manufactured parts (Hong-Seok and Mani, 2014), identified

2.3 Deep learning approaches

In the future, these expert competencies can be replaced with the help of autonomous and data-based decision-making processes. A significant contribution is made by novel DL-based approaches that provide advanced geometry analysis methods to support the RE process beneficially (Guo *et al.*, 2021). The basis for this is formed by geometry-related artificial neural network architectures adapted to the respective geometric data representation.

This is reflected in the wide range of DL applications in the design pipeline that involve similar subtasks of RE in a geometry analysis manner. For example, machining feature recognition of chamfers, holes,

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

and pockets are being discussed (Takashima and Kanai, 2021), which is intended to enable more integrated computer-aided process planning and manufacturing (Shi *et al.*, 2022; Zhang *et al.*, 2018). Particularly in additive manufacturing, DL applications provide advanced geometric information on the technical component (Wong *et al.*, 2020). For example, geometric reasoning regarding thermal deviation or comprehensive manufacturing parameters such as build time can be applied, allowing for a more precise specification of the model geometry. Often 3D convolutional neural networks (3D-CNN) are used for this purpose because the spatial structure of the learned kernels can directly map neighbor relationships in a voxel field, and the data preprocessing is similar for different use cases. This multi-stage approach of repeated convolution and pooling layers has enabled recognition accuracies that were considered impossible before, especially in computer vision.

Recently supervised fittings methods, which infer point-wise properties such as segmentation labels of ruled primitive types, are on the rise (Li *et al.*, 2019; Ping *et al.*, 2020; Sharma *et al.*, 2020; Yan *et al.*, 2021). The geometry parameters result automatically, and no further geometry fitting is required. However, the maximum number of 24 primitive instances that can be processed (Li *et al.*, 2019) is a disadvantage, which makes the realistic processing of technical components difficult. (Sharma *et al.*, 2020) extends the application to freeform surfaces. Due to the number of vertices of the point cloud processed so far, the transfer to a realistic use case, with typically more than 100 thousand vertices or more, is still pending.

3 APPROACH USED AND METHODS

Because of the advantageous regular grid-like structure, this study focuses on processing voxel fields by 3D CNNs. The advantage is the simultaneous processing of global contextual features of the geometric model and their local neighborhood relations of an observations volume simultaneously. Although captured triangle meshes must be transformed into a voxel field, as the 3D rasterization of the captured triangle mesh, the referencing and back-projecting of the predictions onto the triangle mesh cannot be bypassed in any DL approach. The voxel field section, which is input to the 3D CNN network, is hereafter referred to as the observation tile, having a specific resolution defined by the edge length of 32 voxels.

Also, the common problem of the aliasing effect of rasterized data formats shall be avoided using a signed distance field (SDF) representation as the learning input. An SDF is created by computing the continuous signed distance function $dist(p)$ for all points p of the voxel field, where q denotes the closest point on the zero-crossing isosurface:

$$dist(p) = sign(p) * \min(|p - q|) \tag{1}$$

It is explicitly distinguished whether a point p is part of the interior points Ω of the body.

$$sign(p) = \begin{cases} -1 & if\ p \in \Omega \\ 1 & else \end{cases} \tag{2}$$

The implicit SDF geometry representation allows sub-voxel accuracy so that even fine-grained surface structures can be represented, which otherwise would disappear in a typical binary voxel field.

In contrast to current approaches of instance segmentation creating bounding boxes, region masks, or embedded clustering, an explicit per voxel classification is used, where subsurface boundaries are imprinted explicitly in the semantic labels. The hypothesis is that, unlike 2D processing, where the mutual instance occlusion impedes the belonging of an individual pixel to a semantic instance in image space, 3D space occlusion does not exist. Therefore, no further assumptions and processing steps of the underlying network structure should be necessary.

Further, it is assumed that the local surface information of adjacent subsurfaces is sufficient to distinguish between instances of the same geometric type. The primitives to be recognized are always smooth subsurfaces of geometric continuity (GC). Therefore, even for tangential surface transitions, the GC^2 discontinuity between different subsurfaces should be sufficient to keep these subsurface instances apart. For this reason, the data annotation for the semantic segmentation task imprints each potential subsurface region explicitly by wrapping each instance with classified boundary voxels. The shape of the boundary classification corresponds to model feature lines and thereby separates subsurface instances of the same class and other class types.

For modeling most technical components, small subsurfaces can be decisive in creating valid and watertight boundary representations (B-rep). Therefore, the captured mesh will be analyzed piece by

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

piece, similar to a sliding windows algorithm. As already explained, this is based on continuous sub-surfaces being defined locally. Even small observation tiles should lead to a correct surface classification as long as the learned network is sensitive enough. The individual window positions in the sliding window algorithm are determined by a hierarchical structure of an octree, which is also part of the SDF creation. Since the padding mechanism in CNN architectures results in slightly degraded inference accuracies, these border effects are circumvented by an additional overlap of the window dimensions.
After the semantic segmentation on the technical component, the individual voxels predictions need to be consolidated. For this purpose, a region aggregation will be applied using simple region growing to create potential candidates for coherent subsurface regions, which will be called candidate regions.

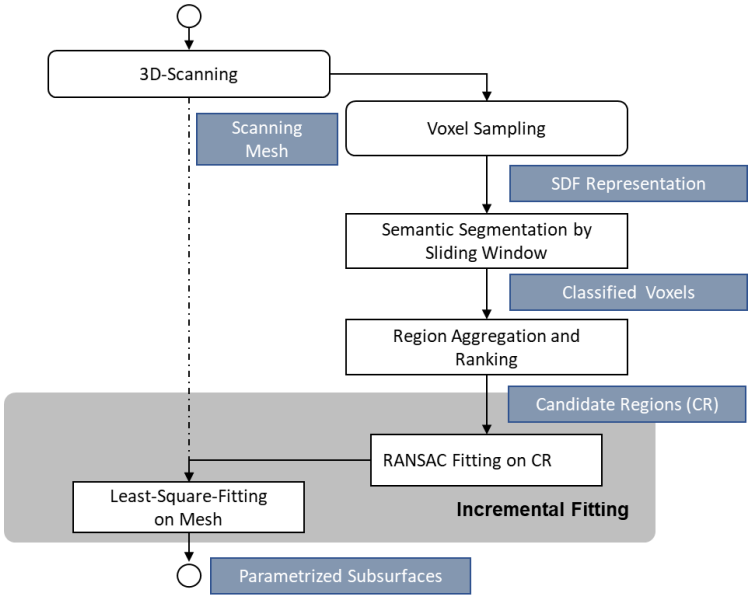

Figure 2. Overall processing pipeline for subsurface detection and parametrization

These region candidates should fundamentally correspond to the subsurface instances to be detected. Finally, the results have to be back-projected to the actual mesh. A two-stage fitting of the segmentation will be performed, wherein a RANSAC is first conducted on the points of the sampled SDF triangles to obtain an initial parametrization of the subsurface parameters. In the second step, an exact parameter fitting is conducted employing a least-square fitting in the narrowband of the initial parameterization on all points of the scanning mesh to obtain the final corresponding triangles belonging to the respective sub-surfaces.

4 METHOD AND IMPLEMENTATION DETAILS

4.1 Synthetic data generation

Since no real-world datasets contain sufficient acquisition data from different scanning methods, a synthetic dataset is created to label, modify, and generate the training instances automatically. For this reason, an adapted data generation pipeline is set up, starting from the B-rep models of the ABC dataset as nominal geometries to create the labeled training instances as SDF voxel tiles (see Figure 3). Seven different types of ruled geometric primitives (plane, sphere, cylinder, conus, and torus) are included in the training dataset. Each selected component must contain all primitive classes at least once, so no further data balancing is required. Even if the models are categorized in the ABC dataset, it is observed that some components in the dataset are similar in their characteristics or represent the same geometry. Therefore, to ensure a wide diversity of variation and to avoid repetitive processing of the same design features that do not provide additional information for the learning process, all but one component with the same file size is deleted from the dataset. The primitive class of the subsurface and its instance label is obtained directly from the B-rep and transferred to a triangular mesh representation. Nevertheless, some sub-surfaces must be merged by checking the equality of the B-rep parameterization since they correspond to the same instance.

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

In contrast to other approaches like (Li *et al.*, 2019) and (Sharma *et al.*, 2020), which modify the nominal mesh exclusively via white noise to overcome the domain gap between synthetic and real-world geometry data, a more sophisticated approach of geometry deformation is followed here, which imprints patterns, surface structures and defects on the components nominal surface. For this purpose, various runs of a mesh deformation are applied, where the mesh vertices are translated at their area-weighted normal direction. To simulate real-world surface deviation and to account for a broad scope of geometric patterns, 3D fractal, cubic and cellular noise are overlayed to morphe a refined mesh. The deformed surface requirements are controlled by the mesh refinement, in which the geometry resolution and resulting discretization error are controlled through a defined deviation tolerance. The deviation tolerance is adaptively controlled to avoid aliasing effects by choosing the resulting point density smaller than the voxel size used later in the observation tile to prevent undersampling. Like the Nyquist frequency, this deviation tolerance in the meshing step is adjusted to be at least two times smaller than the smallest voxel size on a per-model basis. Furthermore, the maximum allowed deviation in mesh deformation is determined based on tolerance class A for surface profiles of (*DIN 2769*), which defines standardized tolerance values for geometric products. All triangles containing at least one vertex outside the tolerance class A are annotated with the outlier class.

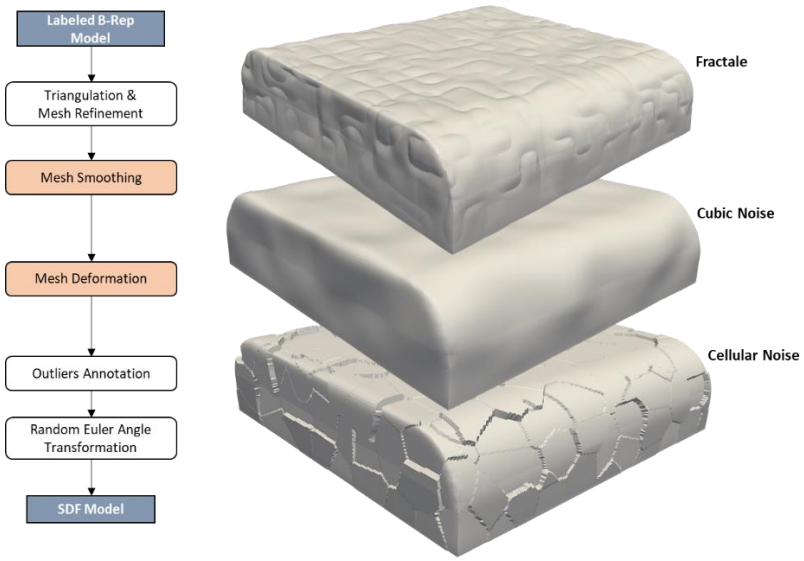

Figure 3. Synthetic data generation and augmentation pipeline

Before sampling the mesh to the SDF representation, all components are rotated at random Euler angles to increase the transformation invariance. In voxelization, the sampling rates range from 80 to 240 voxels in the most extensive space direction per model with an increment of 20 voxels, resulting in an overall dataset size of 105.556 observation and label tiles. When creating the observation tile and the related label tile, the SDF values of the observation tile are normalized by dividing by the used voxel size to ensure that the values are independent of the chosen voxel resolution. The mesh labels are linked to the voxels by transferring the nearest triangle label to the respective voxel represented as the one hot representation. This only accounts for the voxels in the narrowband of the isosurface. All other voxels with values <-1 are labeled as the interior class, and >1 are labeled as the exterior class.

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

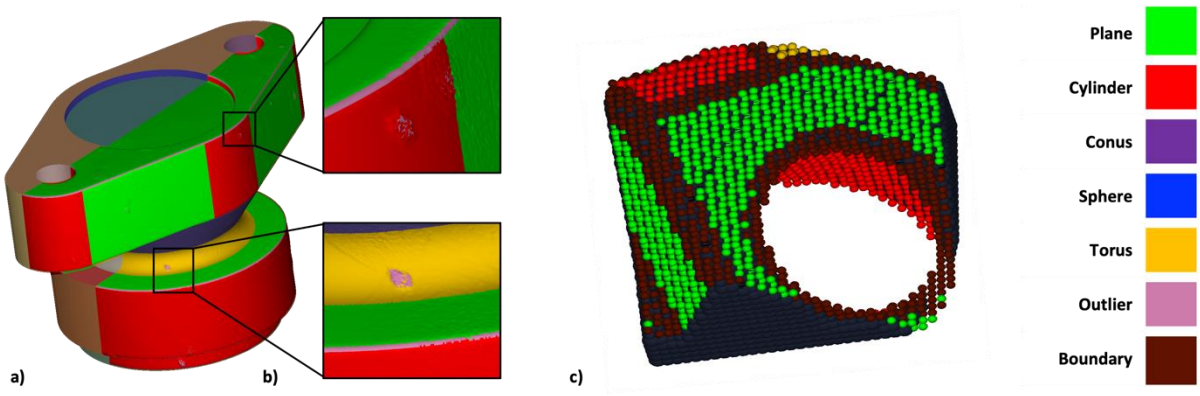

Figure 4. a) Random colored nominal mesh. b) Morphed mesh with colored primitives and outliers. c) Label tile with boundary labels

Unlike the other label classes, boundary classification between subsurface instances is not introduced until the label tile is created. For this purpose, it is checked whether there is a change of instance state in the 16-space of the respective voxel. If there is a change of instance state, the voxel is labeled with the boundary label, not the original primitive class label. Only outlier class voxels remain constant because these are usually minor regions in the voxelized representation and would vanish completely. This annotation ensures that the instance boundaries are always mapped in the exact dimensioning in the voxel tile. For comparability of the SDF approach, the whole training dataset observation tile is converted to the binary case, so it can be investigated whether the implicit SDF representation significantly increases the prediction performance.

4.2 Network architecture

For the supervised deep learning phase, the VoxSegNet architecture (Wang and Lu, 2018) is applied, which satisfies various requirements for the given semantic segmentation task. It is based on the atrous convolutional operation to address the problematic spatial downsampling of pooling layers in CNN models by employing a spacing between the processed values of the convolutional kernel at different rates. In VoxSegNet, the input features are encoded at the dual so-called spatial dense extraction stage, consisting of stacked atrous residual blocks (ARB). These ARB layers yield a multi-scale extraction of high-level semantics, thus reducing the computing complexity and generating specific activation maps. The so-called attention feature aggregation block (AFA) converts the activation dimension to a relevant feature representation in a semantic segmentation manner. Only the softmax layer is modified to the needed class count. Further details of the detailed architecture VoxSegNet can be found in (Wang and Lu, 2018).

4.3 Evaluation metrics and network training

In the learning phase, different evaluation metrics are utilized to benchmark the detection rates of the network architectures in terms of ground truth similarity based on true/false positives (*TP/FP*) and true/false negatives (*TN/FN*). The intersection over union (*IoU*) accounts for an insensitive variation metric, while the dice score (*DS*) applies a more outlier robust metric. The average Hausdorff distance (*AHD*) provides an outlier-sensitive metric (Aydin *et al.*, 2021), where 0.0 is the optimum as the prediction matches the ground truth.

To accelerate the supervised learning process and stabilize the gradient descent, a cross-entropy loss function was applied with a batch size of 4 with a decayed learning rate of *0.0001* and the adaptive moment estimates (ADAM), with a learning rate decay per epoch of $e^{0.03}$ and sparse categorical cross-entropy loss.

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

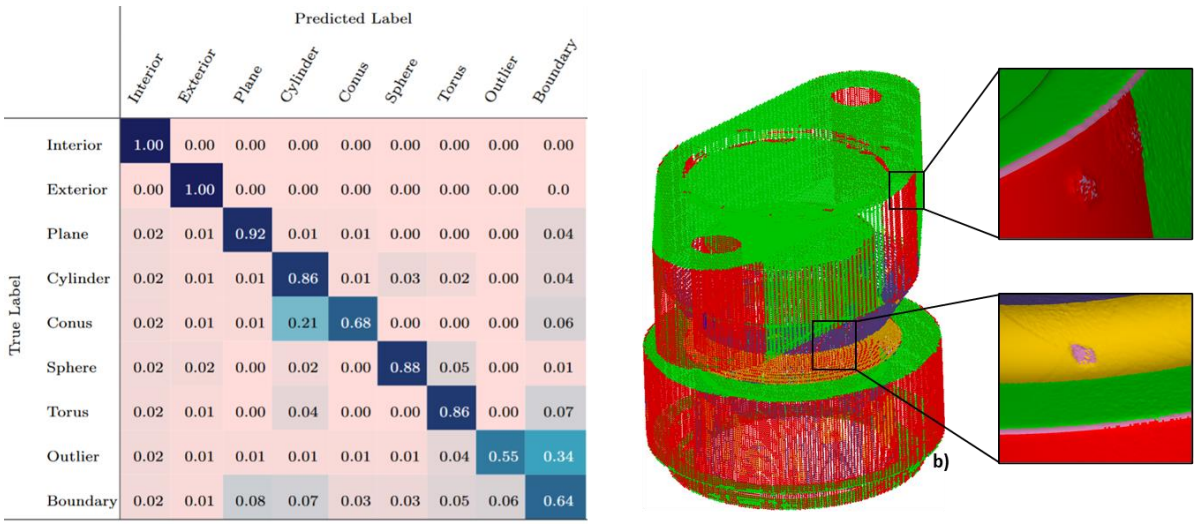

Figure 5. Per-class confusion matrix after 200 epochs on the test set.

After 200 epochs of learning on the synthetic dataset and reaching an *mIOU* of 74.2% and a *DS* of *88.73%* on the test set, it can be concluded that the network model generalizes to extend the geometric classification scheme with the outlier and boundary classes. In multi-class adjacent areas, distinct fine-grained features and accurate region boundaries at instance transitions can be observed. Also, the observed *AHD* of *1.83* resembles an outlier robust segmentation. By comparing the results with a confusion matrix, the prediction rates and inter-class dependencies of all different class types become evident.

A particular correlation can be observed between cylindrical and conical predictions, where there is an accumulation of *FN* of incorrectly predicted coni types. The similarity of both classes can explain this if the conical frustums minor and major radius are nearly the same. Nevertheless, the results of all ruled primitive types achieve a *TP* rate higher than 68% for the worst case. Especially for the planar type, the highest *mIOU* of 93.2% is achieved. An adverse misclassification exists for the additionally introduced outlier and boundary class, which leads to incorrect prediction errors. The outlier voxels are often assigned as boundary voxels with an *FN* of 0.34% at the class transition to the boundary case. This is not considered disadvantageous concerning the overall process since primitive-type classifications are not affected.

5 RESULTS AND DISCUSSION

After the learning phase has been conducted, the overall defined process for subsurface detection and parametrization is demonstrated in a use case (Figure 6). For this purpose, an additive manufactured component is 3D-scanned and digitized. Afterwards, the process steps of semantic segmentation by the trained 3D-CNN, the incremental fitting on the aggregated candidate regions is carried out to parametrize the individual primitive instances without additional human supervision.

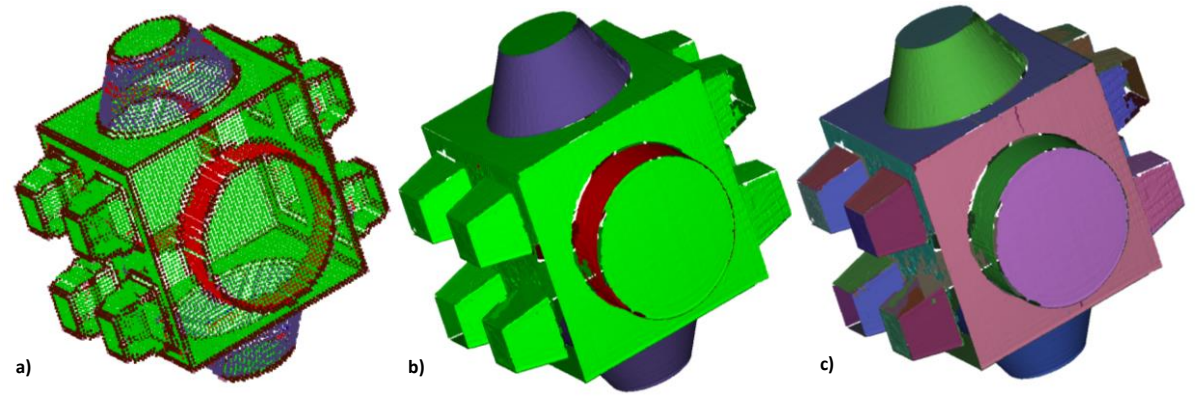

Figure 6. a) Inferenced semantic segmentation. b) Fitted primitives, colored by type. c) Primitive instances, randomly colored.

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

As a result, valid and robust surface detections of all relevant primitive classes could be achieved, so that accurate surface parameterizations are derived despite the manufacturing inaccuracies of the additive manufactured component. The introduced boundary labels allow the separation of individual subsurface instances. However, it remains to be examined whether the detection rate is reduced in the case of smooth subsurface transitions because sharp labels are often the more distinguishable feature. Currently, the intended inference of defect regions with the outlier class has been limited to rounded subsurface border regions. It is assumed that the boundary labels have often obscured the outlier voxels, so an improvement labeling strategy needs to be established in the future.
Nevertheless, with the introduction of the additional label classes into the learning phase, a distinct segmentation result was obtained. This could lead to more accurate parametrization results in the future since identified outlier regions are not part of the fitted surface regions. In particular, it can be emphasized that despite the additional outlier category, the other geometry knowledge available does not produce any degradation of the primitive detection accuracy compared to other approaches.

6 CONCLUSION AND OUTLOOK

In the context of a more robust surface reconstruction to outlier and noisy acquisition data, the increasingly common deep learning approaches offer the possibility to integrate a deeper geometry model understanding into the RE process pipeline. Therefore, in this research paper, an enhanced method was proposed that combines tolerable surface deviations and surface defects directly into the process step of geometry detection, deriving subsurface instances of ruled geometries by an explicit data labeling of instance boundaries. For this purpose, a 3D-CNN-based geometry recognition is applied for semantic segmentation to detect surface outliers and the subsurface boundaries to separate primitive instances. Using a sophisticatedly created training synthetic dataset, the explicit labeling strategy enables for the separation of adjacent subsurfaces. The utilized SDF representation improves resolution and recognition accuracy and mitigates aliasing effects. In contrast to other network structures with sophisticated network architecture, the approach presented here does not require any architecture adaptation due to the use of a relatively simple network architecture.
A benchmark dataset should be established in the future, providing a more accurate evaluation and comparability baseline with other RE approaches. Further predictions regarding the surface quality need to be integrated into the dataset labeling, directly providing algorithmic parameters required for fitting.

REFERENCES

Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N. and Azariadis, P. (2007), "3D Mesh Segmentation Methodologies for CAD applications", *Computer-Aided Design and Applications*, Vol. 4 No. 6, pp. 827–841. http://doi.org/10.1080/16864360.2007.10738515.

Ali, S., Durupt, A. and Adragna, P.A. (2013), "Reverse Engineering for Manufacturing Approach: Based on the Combination of 3D and Knowledge Information", *Smart Product Engineering*, *Lecture Notes in Production Engineering*, pp. 137–146. http://doi.org/10.1007/978-3-642-30817-8_14.

Attene, M., Falcidieno, B. and Spagnuolo, M. (2006), "Hierarchical mesh segmentation based on fitting primitives", *The Visual Computer*, Vol. 22 No. 3, pp. 181–193. http://doi.org/10.1007/s00371-006-0375-x.

Aydin, O.U., Taha, A.A., Hilbert, A., Khalil, A.A., Galinovic, I., Fiebach, J.B., Frey, D. and Madai, V.I. (2021), "On the usage of average Hausdorff distance for segmentation performance assessment: hidden error when used for ranking", *European radiology experimental*, Vol. 5 No. 1, p. 4. http://doi.org/10.1186/s41747-020-00200-2.

Bici, M., Mohammadi, S.S. and Campana, F. (2020), "A Compared Approach on How Deep Learning May Support Reverse Engineering for Tolerance Inspection", *Proceedings of the ASME International Mechanical Engineering Congress and Exposition - 2019, November 8-14, 2019, Salt Lake City, Utah, USA*. http://doi.org/10.1115/IMECE2019-11325.

Buonamici, F., Carfagni, M., Furferi, R., Governi, L., Lapini, A. and Volpe, Y. (2018), "Reverse engineering modeling methods and tools: a survey", *Computer-Aided Design and Applications*, Vol. 15 No. 3, pp. 443–464. http://doi.org/10.1080/16864360.2017.1397894.

Chikofsky, E.J. and Cross II, J.H. (1990), "Reverse engineering and design recovery: a taxonomy", *IEEE Software*, Vol. 7 No. 1, pp. 13–17. http://doi.org/10.1109/52.43044.

DIN 2769: Toleranzen für Längen - und Winkelgrößenmaße ohne individuelle Toleranzangaben No. 2769, Beuth Verlag GmbH, Berlin.

https://doi.org/10.1017/pds.2023.178 Published online by Cambridge University Press

Fayolle, P.-A. and Pasko, A. (2016), "An evolutionary approach to the extraction of object construction trees from 3D point clouds", *Computer-Aided Design*, Vol. 74, pp. 1–17. http://doi.org/10.1016/j.cad.2016.01.001.
Geng, Z., Sabbaghi, A. and Bidanda, B. (2022), "A framework of tolerance specification for freeform point clouds and capability analysis for reverse engineering processes", *International Journal of Production Research*, Vol. 60 No. 24, pp. 7475–7491. http://doi.org/10.1080/00207543.2022.2086083.
Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. and Bennamoun, M. (2021), "Deep Learning for 3D Point Clouds: A Survey", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 43 No. 12, pp. 4338–4364. http://doi.org/10.1109/TPAMI.2020.3005434.
Hong-Seok, P. and Mani, T.U. (2014), "Development of an Inspection System for Defect Detection in Pressed Parts Using Laser Scanned Data", *Procedia Engineering*, Vol. 69, pp. 931–936. http://doi.org/10.1016/j.proeng.2014.03.072.
Kaisarlis, G.J., Diplaris, S.C. and Sfantsikopoulos, M.M. (2008), "Geometrical position tolerance assignment in reverse engineering", *International Journal of Computer Integrated Manufacturing*, Vol. 21 No. 1, pp. 89–96. http://doi.org/10.1080/09511920601164140.
Kaiser, A., Ybanez Zepeda, J.A. and Boubekeur, T. (2019), "A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data", *Computer Graphics Forum*, Vol. 38 No. 1, pp. 167–196. http://doi.org/10.1111/cgf.13451.
Li, L., Sung, M., Dubrovina, A., Yi, L. and Guibas, L. (2019), "Supervised Fitting of Geometric Primitives to 3D Point Clouds", pp. 2647–2655. http://doi.org/10.1109/CVPR.2019.00276.
Ping, G., Esfahani, M.A. and Wang, H. (2020), "Unsupervised 3D Primitive Shape Detection using Mathematical Models", 2020 *16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 178–183. http://doi.org/10.1109/ICARCV50220.2020.9305494.
Qie, Y., Bickel, S., Wartzack, S., Schleich, B. and Anwer, N. (2021), "A function-oriented surface reconstruction framework for reverse engineering", *CIRP Annals*, Vol. 70 No. 1, pp. 135–138. http://doi.org/10.1016/j.cirp.2021.04.016.
Schnabel, R., Wahl, R. and Klein, R. (2007), "Efficient RANSAC for Point-Cloud Shape Detection", *Computer Graphics Forum*, Volume 26, Issue 2, pp. 214–226. http://doi.org/10.1111/j.1467-8659.2007.01016.x.
Schöne, C. (2009), "Reverse Engineering für Freiformflächen in Prozessketten der Produktionstechnik", Habilitation, Technische Universität Dresden, Dresden, 2009.
Shah, G.A., Polette, A., Pernot, J.-P., Giannini, F. and Monti, M. (2022), "User-Driven Computer-Assisted Reverse Engineering of Editable CAD Assembly Models", *Journal of Computing and Information Science in Engineering*, Vol. 22 No. 2. http://doi.org/10.1115/1.4053150.
Sharma, G., Liu, D., Maji, S., Kalogerakis, E., Chaudhuri, S. and Měch, R. (2020), "ParSeNet: A Parametric Surface Fitting Network for 3D Point Clouds", *Computer Vision – ECCV* 2020, *Lecture Notes in Computer Science*, Vol. 12352, pp. 261–276. http://doi.org/10.1007/978-3-030-58571-6_16.
Shi, P., Qi, Q., Qin, Y., Scott, P.J. and Jiang, X. (2022), "Highly interacting machining feature recognition via small sample learning", *Robotics and Computer-Integrated Manufacturing*, Vol. 73, p. 102260. http://doi.org/10.1016/j.rcim.2021.102260.
Takashima, H. and Kanai, S. (2021), "Recognition of Free-form Features for Finite Element Meshing using Deep Learning", *Computer-Aided Design and Applications*, Vol. 19 No. 4, pp. 677–693. http://doi.org/10.14733/cadaps.2022.677-693.
Várady, T., Martin, R.R. and Cox, J. (1997), "Reverse engineering of geometric models—an introduction", *Computer-Aided Design*, Vol. 29 No. 4, pp. 255–268. http://doi.org/10.1016/S0010-4485(96)00054-1.
Wang, Z. and Lu, F. (2018), *VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes*, available at: http://arxiv.org/pdf/1809.00226v1.
Wong, V.W.H., Ferguson, M., Law, K.H., Lee, Y.-T.T. and Witherell, P. (2020), "Automatic Volumetric Segmentation of Additive Manufacturing Defects with 3D U-Net", *AAAI 2020 Spring Symposia*. http://doi.org/.
Woodford, O.J., Pham, M.-T., Maki, A., Perbet, F. and Stenger, B. (2014), "Demisting the Hough Transform for 3D Shape Recognition and Registration", *International Journal of Computer Vision*, Vol. 106 No. 3, pp. 332–341. http://doi.org/10.1007/s11263-013-0623-2.
Xie, Y., Tian, J. and Zhu, X.X. (2020), "Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation", *IEEE Geoscience and Remote Sensing Magazine*, Vol. 8 No. 4, pp. 38–59. http://doi.org/10.1109/MGRS.2019.2937630.
Yan, S., Yang, Z., Ma, C., Huang, H., Vouga, E. and Huang, Q. (2021), *HPNet: Deep Primitive Segmentation Using Hybrid Representations*, available at: http://arxiv.org/pdf/2105.10620v3.
Zhang, Z., Jaiswal, P. and Rai, R. (2018), "FeatureNet: Machining feature recognition based on 3D Convolution Neural Network", *Computer-Aided Design*, Vol. 101, pp. 12–22. http://doi.org/10.1016/j.cad.2018.03.006.