




AUTOMATED PART DECOMPOSITION FOR PRODUCT ARCHITECTURE MODELING

J. Redeker , P. Gebhardt, A.-K. Reichler, E. Türck, K. Dröder and T. Vietor

Technische Universität Braunschweig, Germany

 j.redeker@tu-braunschweig.de

Abstract

This paper presents an algorithm that contributes to an automatic decomposition of a mechanical part based on geometric features and methods of unsupervised machine learning. For the development of the algorithm, existing techniques of 3D shape segmentation, especially surface-based part segmentation procedures are reviewed and important areas of activities are revealed. The developed multi-step approach results in an abstract product model. This representation leads to a new way of designing and redesigning parts for the novel hybrid manufacturing concept Incremental Manufacturing (IM).

Keywords: design automation, conceptual design, knowledge-based engineering (KBE), 3D shape segmentation, additive manufacturing

1. Introduction

The growing demand for technical-functional differentiation, as well as customized products, leads to an increasing number of product variants while the batch size decreases. For smaller batch sizes, the economic efficiency of conventional processes decreases, caused by higher variant-specific investments and batch-specific set-up costs. Therefore, new manufacturing approaches are required, which allow a reduction in variant-specific expenses, but at the same time, guarantee sufficient production output capability.

A novel manufacturing approach that faces these requirements is Incremental Manufacturing (IM). The IM approach consists of a production process where products are built up from pre-produced parts which are finalized by additive, subtractive, or assembly production steps, called increments. This manufacturing approach enables a cost-efficient production of customized multi-material parts (Dröder et al., 2016, 2017). The free combination of different incremental steps increases the decision-making freedom in the part design and production planning phase. However, the design specifications have a more restrictive effect on the production route than established concepts.

To reduce the arising complexity from the new design freedom of IM, a process model was developed by Reichler et al. (2019). This model, which is briefly described in section 1.1, contributes to the automation of the redesigning process of parts for IM. A central starting point for the redesign is the algorithm-based decomposition of a reference part. Thus, this paper presents a novel shape segmentation approach, which contributes to the automation of the decomposition process using a density-based cluster algorithm.

Section 1 describes the main idea of the integrated design and process planning approach for IM as well as the scope of this paper is given. Section 2 gives an overview of the topic of shape analysis with a focus on shape segmentation techniques. The main topics are part-based mesh segmentation and surface-

based mesh segmentation approaches. Section 3 addresses the results from the previous considerations and defines requirements for the segmentation algorithm based on the related work. Furthermore, the developed algorithm is described in detail as well as further applications are pointed out. By applying the algorithm to a choice of benchmark shapes, the capability of the segmentation approach is demonstrated. After a summary of the results, section 4 is pointing out further areas of activity and subsequent steps.

1.1. Design and process planning for incremental manufacturing

A procedure model was developed by Reichler et al. (2019) to handle the large solution space created by the novel manufacturing concept of IM. This model contributes to the reduction of the theoretically conceivable degree of freedom in product and process design to practicable and variant-flexible combinations in the early design phase. Besides, the integrative approach enables an equal optimization of part properties and the manufacturing sequences during the redesign process of a product. Figure 1 shows the developed procedure model for the integrated part design for IM.

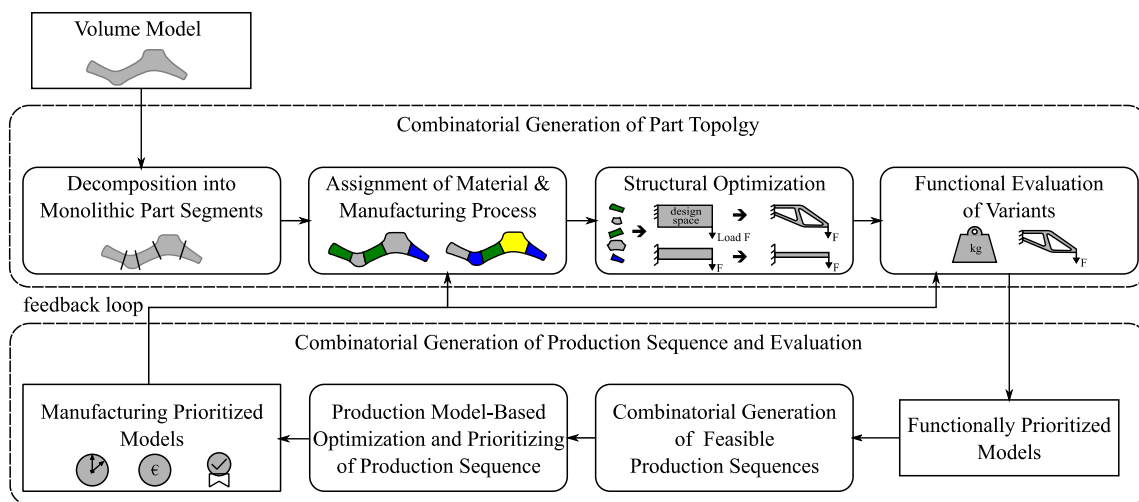


Figure 1. Procedure model for the integrated part design for IM acc. to Reichler et al. (2019)

The model is categorized into the main steps “combinatorial generation of the part topology” and “combinatorial generation of production sequence and evaluation”. These steps are carried out alternately and are mutually refined by a feedback loop. The method starts with the decomposition of a digital model of a reference part into segments to be manufactured monolithically. Materials along with manufacturing strategies (pre-produced part, additive, subtractive, etc.) are then allocated by previously defined heuristics (e.g., material compatibility, qualification of joining technology, mechanical load, etc.) to the specified segments. The allocation of different materials and production strategies leads to a tree-like structure of part variants. These variants are designed and sized using methods of structural optimization (topology and parameter optimization). Following, the resulting part candidates are ranked concerning properties like weight, stiffness, and cost. Based on the functional evaluation of the design solutions, a combinatorial generation of possible manufacturing sequences is carried out in the second main step of the method. Afterwards, the determined sequences are optimized and evaluated concerning key manufacturing figures. Finally, the prioritization leads to a ranking of different manufacturing sequences based on production time, cost, and quality. The extracted manufacturing knowledge is transferred to the first step and used here, especially for the modification and concretization of the part redesign as well as the elimination of variants.

1.2. Scope of this paper

In order to realize a high degree of automation for the entire procedure model and thus shorten the time for the design and process planning for IM, it is necessary to automate the decomposition of the parts to be redesigned. All further steps are based on the successful segmentation of the reference part into meaningful segments, which can be manufactured monolithically. Humans are easily capable to

abstract the shape of an object, to decompose it based on geometric properties, or to group multiple objects based on shape similarities. The automation of these exemplary mentioned tasks, which is still challenging to computers, is dedicated to the topic of shape analysis (Laga et al., 2019).

Firstly, suitable approaches for an automated shape segmentation from the area of shape analysis have to be identified. On this basis, an algorithm has to be developed that enables an automated decomposition of mechanical reference parts into regions that can be manufactured monolithically, and therefore considers the incremental principle of the underlying manufacturing approach. Following the part segmentation, it is useful to transfer the resulting components into an abstract representation e.g. a graph-based product structure. In the context of product architecture design, the product structure represents an important starting point for the redesign process of a product (cf. Richter et al., 2016). Thus, analyzing the reference part on the level of product architecture may enable a more detailed structuring of individual part segments, their grouping, as well as the moving of segment boundaries in the future. Additionally, representing the parts of being redesigned more abstractly (e.g., graph data structure) brings the advantage that it can be numerically analyzed and processed.

2. Related work

This paper deals with the development of an algorithm for automated geometry segmentation based on a digital representation of a reference part. Therefore, this section gives a brief overview of the procedures of 3D shape segmentation, which is a topic of the area of shape analysis.

2.1. Part-based mesh segmentation

Shape segmentation is a fundamental research topic in shape analysis and aims to decompose a digital shape into subparts. Since many of the methods are based on meshes featuring the 3D object, the topic is also referred to as 3D mesh segmentation. In general, mesh segmentation algorithms can be divided into the categories of chart-based and part-based segmentation techniques (Rodrigues et al., 2018).

In the context of this paper, part-based segmentation plays the most important role. The focus of part-based segmentation techniques is on decomposing the reference geometries into meaningful parts, which are consistent with user intention, geometric mesh attributes, and human shape perception. Whereby the corresponding application strongly influences the meaningfulness. A general requirement for a good segmentation algorithm is that the properties of the mesh elements within a segment should have high similarity. Furthermore, the association between different regions should be low, and the segment boundary should be geometrically tight and smooth. Finally, the segment boundary should match human perception and should reflect significant features, i.e., sharp edges or corners (Zhang et al., 2012).

In their comprehensive survey on the properties and limitations of mesh segmentation algorithms, Rodrigues et al. (2018) point out that most state-of-the-art frameworks take advantage of the minima rule proposed by Hoffman and Richards (1984), to achieve a boundary that matches human perception. The core statement of the rule is that humans perform shape decompositions at negative minima of principal curvatures. Segment boundaries are thus drawn in concave areas.

Furthermore, Rodrigues et al. (2018) define three main categories of part-based segmentation techniques. In particular, volume-based segmentation, skeleton-based segmentation, and surface-based segmentation.

In **volume-based segmentation**, the input is a 3D volumetric mesh (e.g. tetrahedral) that is partitioned into volumetric sub-meshes. As a result of their survey Rodrigues et al. (2018) note that most of the methods in this category are not automated. This means that the method is not capable to find the number of segments beforehand, or it requires user interaction during the process of volume decomposition.

In **skeleton-based segmentation**, the segments are line segments. The input is either a surface mesh embedded in 3D or a volumetric mesh, but the output is a 1D skeleton that represents the structural shape of the mesh. Therefore, they are less suitable for mechanical parts and are not able to capture small features. In contrast, these methods are well suited for tubular freeform objects.

In **surface-based segmentation**, the segments represent 2D sub-meshes or regions of a 2D triangular mesh, where each region consists of a set of connected facets that have similar geometric properties. Similar to volume-based segmentation, the main issue has to do with the lack of an automated procedure to determine the number of segments before decomposing a mesh into several meaningful regions. Depending on the application, a surface-based technique can be used to achieve other types of

segmentation, i.e., volume-based segmentation, or skeleton-based segmentation. This is due to the fact that a watertight surface mesh representing a 3D object also provides volumetric information. In summary, the surface-based methods seem to offer the highest flexibility and suitability in terms of an automated shape segmentation. Besides, the literature shows that this category of segmentation methods provides a wider variety of approaches compared to the other reviewed techniques. Thus, the different procedures in the area of surface-based methods are examined in section 2.2 in more detail.

2.2. Surface-based segmentation

An evaluation of the surveys on mesh segmentation by [Agathos et al. \(2007\)](#), [Shamir \(2008\)](#), [Theologou et al. \(2015\)](#) and [Rodrigues et al. \(2018\)](#) shows that the surface-based segmentation approaches can be categorized into four major groups. Namely, region growing, watershed-based segmentation, cluster analysis (iterative and hierarchical clustering), and boundary-based segmentation (implicit methods).

The methods of **region growing** decompose a mesh into polygonal regions that are not strictly convex. The basic idea behind the region growing algorithms is that each region starts with a random or predefined seed vertex or element of the mesh to be decomposed and increases in the number of clustered polygons. The region keeps growing as long as a specified condition (e.g., a convexity-related condition) is fulfilled. Currently, there appears to be no published approach that can determine the number of region seeds and region number beforehand. As a consequence, the segmentation results may be unnatural ([Rodrigues et al., 2018](#)).

In the category of **watershed-based** algorithms, the segmentation is achieved in analogy to the way water fills a geographic surface. If several water sources fill the surface basins, points are created at the flood regions meeting point. These points construct the watershed lines which are used to decompose the surface into segments ([Agathos et al., 2007](#)). Since the underlying procedure has strong analogies to region growing, there are similar disadvantages. For this reason, the algorithms are only conditionally suitable for a meaningful automated segmentation.

The **boundary-based** segmentation approaches attempt to locate the boundaries of the segments rather than the segments themselves. This approach assumes that edges that lie on the boundary of two segments have similar geometric attributes ([Theologou et al., 2015](#)). The challenge with boundary-based methods is that there can be several boundaries that divide two regions. Thus, a weighting for the most relevant ones has to be performed. Furthermore, it is not always possible to find a closed boundary or loop of concave edges separating two regions [Rodrigues et al. \(2018\)](#).

The methods based on **cluster analysis** rely on iterative and hierarchical cluster algorithms. The basic idea of using cluster algorithms to solve the problem of mesh segmentation is to assign each face or mesh vertex to the (geometrical) nearest segment determined by clustering. A common approach is defining weights between adjacent nodes and calculating the distance of each node to a predefined set of seeds or representatives of the final cluster ([Theologou et al., 2015](#)). The central issue in iterative clustering-based methods, which are in general based on some k-means clustering procedure, is to determine the number of clusters and thus segments a priori. This is due to the nature of the k-means algorithm, which requires the number of clusters as input parameters. Regarding the hierarchical clustering-based methods, it seems that no method automatically calculates the number of regions or clusters a priori [Rodrigues et al. \(2018\)](#).

In summary, it appears that only a few state-of-the-art procedures allow a fully automated segmentation of mechanical parts in a meaningful way. Thus, [Rodrigues et al. \(2018\)](#) also identify the a priori calculation of the number of mesh segments in the context of a high degree of automation as an essential research topic for the future. A large number of cluster-based methods achieve satisfying results in shape segmentation that matches either human perception about the shape or the application requirements in a particular context.

Furthermore, the literature shows a wide variety of approaches for extracting the input data based on geometrical attributes of the surface mesh required for the following cluster procedures. The current weakness of this group of algorithms seems to be due to the used iterative and hierarchical clustering approaches, which leads to a reduction in the degree of automation. For this reason, section 3 presents a segmentation approach that relies on a density-based cluster algorithm, which seems to be unique in the context of two-manifold mesh segmentation algorithms. This approach could overcome the

disadvantage that the user has to estimate the number of clusters or segments a priori, which may result in a higher degree of automation in shape segmentation.

3. Algorithm-based decomposition of a reference part

This section introduces the approach of the part-based segmentation algorithm. The review of the related work shows that the group of surface-based segmentation approaches, especially methods based on cluster analysis, tend to be suitable for the automated decomposition of a mechanical reference part. To overcome the issue that only a few state-of-the-art algorithms seem to be capable of a fully automated part segmentation, we present a novel approach that could contribute to a higher degree of automation in 3D shape segmentation. Based on the previous considerations, the following most essential requirements for the segmentation method are defined:

1. Automated surface-based segmentation with as few user-defined input parameters as possible. In particular, the algorithm should be capable of independently determining the number of part segments to be manufactured monolithically.
2. The algorithm must be suitable for the shape segmentation of mechanical parts. These can be characterized by their sharp edges and well-defined surfaces (Koch et al., 2018).
3. Mesh segments within a segment should have high geometrical similarity and should be spatially adjacent. This guarantees a decomposition of the reference part into contiguous regions.
4. The association between different segments should be low, and the segment boundary should be geometrically tight and smooth.
5. Simple to extend with further features (e.g., design feature detection)

3.1. Surface-based segmentation approach

Based on the previously mentioned requirements for the automated decomposition of a reference part into monolithic segments, a surface-based segmentation algorithm is proposed. The procedure can be assigned to the group of cluster analysis-based methods using a graph representation of the input two-manifold mesh (cf. Theologou et al., 2015). The corresponding flow chart of the main procedure is shown in Figure 2. It consists of six phases: (1) taking surface mesh of reference part as input, (2) computation and aggregation of geometrical feature descriptors, (3) assembly of distance matrix, (4) density-based cluster analysis, (5) segmentation of the mesh according to the results of clustering and (6) merging the determined segments based on concavity.

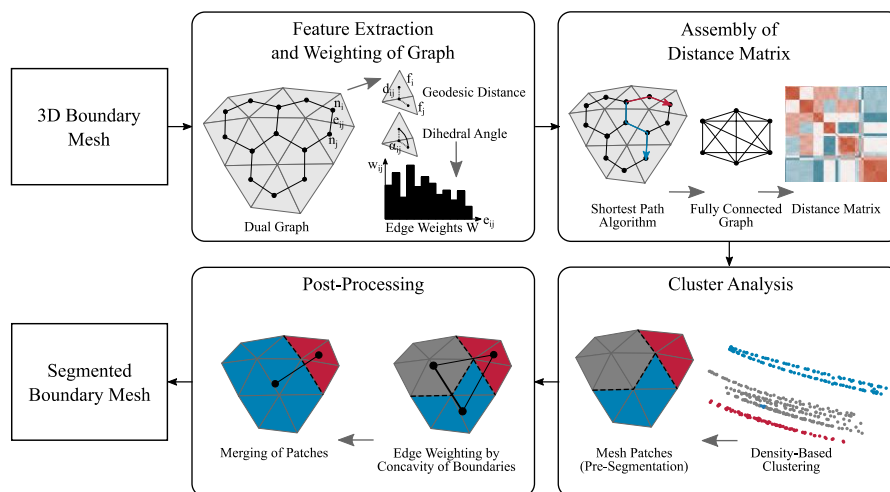


Figure 2. Flow Chart of the proposed surface-based segmentation method

For the segmentation approach, the input surface mesh $M = (V, F)$, defined as a collection of vertices V and faces F , is treated as an undirected graph. Based on this, a weighted dual graph $G = (N, E, W)$ is constructed. The dual-graph is a structure containing a set of nodes $n_i \in N$ and a set of edges $e(n_i, n_j) \in E$ that specifies the connections between the nodes. Each face (triangular element) of F is treated as a node whose

position corresponds to the geometric center (centroid) of the related face $f_i \in F$. Therefore, the edges represent the connections between adjacent faces. The set of weights W is assigned to the edges and describes a distance between the linked nodes. A high geometric similarity corresponds to a small distance.

3.1.1. Feature extraction

In the following, the strength (distance) of the connection between two adjacent nodes n_i and n_j is considered by a weight $w(n_i, n_j) \in W$. For a geometric characterization of the graph, and thus the shape of the part to be segmented, shape descriptors are used. Shape descriptors characterize the geometry, topology, and semantics of 3D objects in a compact numerical representation. They can be distinguished into binary and unary descriptors. Unary descriptors describe the local geometry of a node, whereas binary descriptors represent the geometrical connection between adjacent nodes (Laga et al., 2019).

Common and proven binary descriptors for the partitioning of 3D meshes are the dihedral angle as well as the geodesic distance (Shamir, 2008; Rodrigues et al., 2018). The approach presented in this paper can be extended later by any binary descriptors. An overview of further suitable feature descriptors is given in (Kalogerakis et al., 2010). The dihedral angle α_{ij} between two adjacent faces f_i and f_j is determined by Equation (1).

$$\alpha_{ij} = \cos^{-1} \left(\frac{X_i \cdot X_j}{|X_i| |X_j|} \right) \quad (1)$$

The numerator corresponds to the scalar product of the normal vectors X_i and X_j of the adjacent elements f_i and f_j . The denominator results from the product of the lengths of the vectors (L^2 norm). The dihedral angle also acts as an indicator for concave and convex edges. Whereby an angle greater than π ($\alpha_{ij} > \pi$) indicates a convex edge.

The geodesic distance d_{ij} , being the shortest distance between the centroids of adjacent elements without leaving the surface of M , is determined with Equation (2).

$$d_{ij} = |P_e - P_i| + |P_e - P_j| \quad (2)$$

Here d_{ij} corresponds to the distance between the coordinates of the centroids P_i and P_j of the adjacent faces across the center of the shared mesh element edge P_e (see Figure 2).

Then the set of weights for the characterization of the geometrical strength between adjacent nodes is calculated by the following weighting scheme (Equation 3), inspired by Shlafman et al. (2002):

$$w_{ij} = \gamma(1 - \cos^2(\alpha_{ij})) + (1 - \gamma) d_{ij} \quad (3)$$

This Equation allows a weighting of the binary descriptors, by the factor γ . In our studies, we have set the weighting factor $\gamma = 0.9$, since the dihedral angle usually plays a more dominant role in a geometrically meaningful segmentation (Liu and Zhang, 2004). The expression $(1 - \cos^2(\alpha_{ij}))$ reaches its maximum at $\pi/2$ and its minimum at 0 (or π) and can be seen as an angular distance. Thus, coplanar faces are considered as close to each other.

3.1.2. Assembly of the distance matrix

To get a fully connected graph, i.e., the distance of each node n_i to n_j in G , an implementation of the shortest path algorithm of Dijkstra with Fibonacci heap of the Python library SciPy by Virtanen et al. (2019) is used. The resulting weighted adjacency matrix W of the dual graph G , which is based on the local geometric properties, can now be interpreted as a Euclidean distance matrix. This matrix considers both the spatial and geometric connectivity of the elements of the mesh and should be suitable to detect regions of geometrical similarities of the shape. Furthermore, it contains large distances between adjacent faces with high dihedral angles, and therefore sharp edges are considered.

3.1.3. Cluster analysis

In the following, methods of cluster analysis are used to discover similarities in the predefined distance matrix. Basically, cluster analysis is the process of partitioning a set of data objects into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another. Basic concepts like partitioning (iterative) and hierarchical clustering methods are designed to find

spherical-shaped groups (Han et al., 2012). As highlighted in section 2.1, these algorithms are only conditionally suitable for an automated segmentation process.

To analyze the inherent structure of the distance matrix and to find a suitable cluster algorithm, the data is linearly embedded into the two-dimensional space using a classical multidimensional scaling (MDS). The basic idea behind MDS is to seek a low-dimensional representation of the data in which the distances respect the distances in the original high-dimensional space. The analysis of selected data sets shows a characteristic of non-spherical arrangements of data points with a varying density and connectivity. To discover clusters of arbitrary shape, density-based clustering methods are suitable. These methods also have the characteristic that the number of clusters does not have to be determined a priori (Han et al., 2012)

Therefore, the Ordering Points To Identify the Clustering Structure (OPTICS) cluster algorithm implementation of the Python library Scikit-learn (Pedregosa et al., 2011) is selected. This density-based algorithm is based on the DBSCAN (Density-Based Spatial Clustering for Applications with Noise) algorithm and overcomes the issue that almost all of the well-known clustering algorithms require input parameters, which are hard to determine but have a significant influence on the clustering result. Another advantage is that the number of clusters does not have to be estimated a priori. Furthermore, it is capable of detecting meaningful clusters in data with arbitrary shape and varying density (Ankerst et al., 1999). The only input parameter to be defined is the neighborhood density threshold $MinPts$. In the following studies, the parameter is set to $MinPts = 6$, which is determined based on the element distributions of the input meshes.

At this point, the segmentation results of the algorithm are shown in Figure 3. Whereas Figure 3 a) depicts the obtained clusters. In the following, these clusters are referred to as patches since they do not represent the final segmentation results.

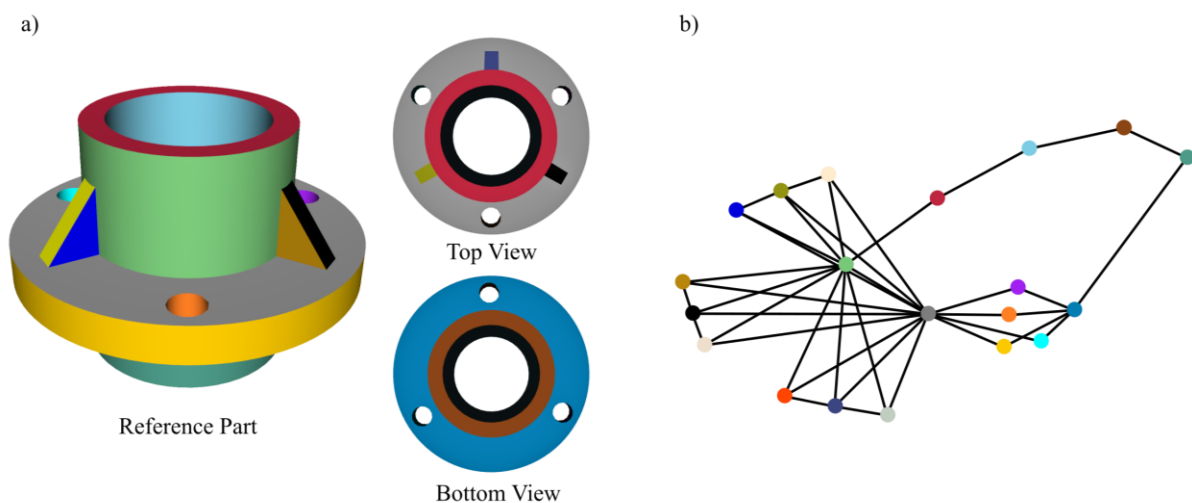


Figure 3. Results of cluster analysis. a) Determined patches by OPTICS, b) Graph of patches

The algorithm clearly detects all 20 boundary surfaces of the reference part. In particular, the figure shows that the detected patches have high geometrical similarity and are spatially adjacent (requirement 3). Furthermore, the patch boundaries are geometrically tight and smooth (requirement 4). Next, the patches and their connections are transferred into a graph-based structure (see Figure 3 b)) using Python library NetworkX (cf. Aric et al., 2008). The coloring of the graph nodes corresponds to the coloring of the clusters. Note that the results of this section seem to be applicable for tasks in design feature recognition in reverse-engineered models, e.g., presented by Gauthier et al. (2019).

3.1.4. Post-processing for final segmentation

To obtain a final segmentation of the pre-segmented part, a post-processing step is added based on the dihedral angles at the patch boundaries and the application of the minima rule (see section 2.1). Thus,

a set of weights C is derived and assigned to the edges of the graph of patches. Each edge weight $c(a_{ij}) \in C$ is defined as follows (Equation 4):

$$c_{ij} = \begin{cases} 0.1 & \text{if boundary is convex} \\ 1 & \text{if boundary is concave} \end{cases} \quad (4)$$

Therefore, the edges of the graph connecting the patches are labeled as convex or concave. These edge attributes lead to the final decomposition based on merging the patches that share a convex edge (see Figure 4 a)). Figure 4 b) shows the related graph, which represents the connections of the final part decomposition. Edges with a greater line width indicate the convex joints. The five determined segments are highlighted in the graph.

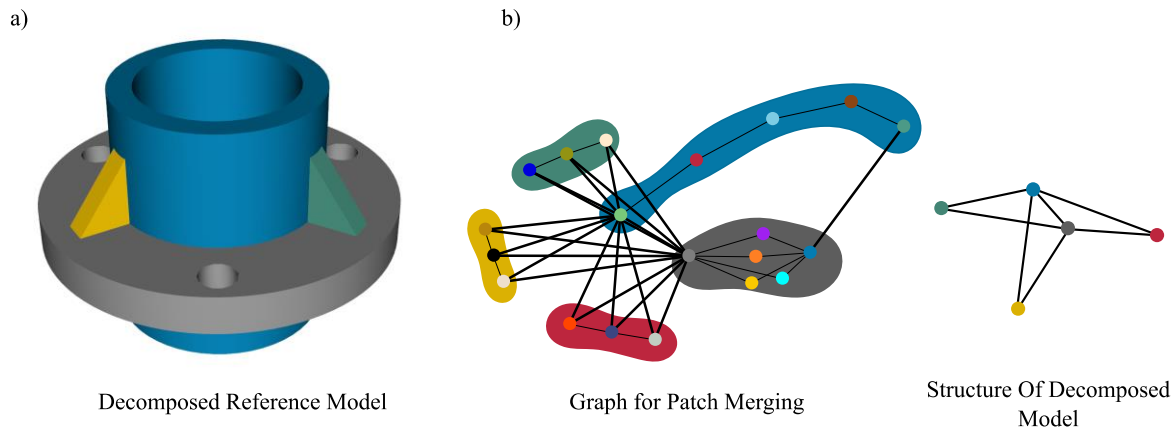


Figure 4. a) Final decomposition of the reference model, b) Resulting graph for patch merging and structure of the decomposed model

The resulting graph is only assigned with geometric attributes (mesh segments) and their physical relations. Hence, it represents a product structure. The nodes and edges of the graph can be interpreted as abstract placeholders and can be filled with arbitrary information (material, manufacturing strategies, joining technology, manufacturing key figures, etc.).

The literature shows that graphs tend to be suitable as a central data model for integrated product and process planning (e.g. [Arnold and Rudolph, 2012](#)). Thus, the automatically generated graph appears to be a promising basis for further steps, like the algorithm-based generation of part variants, in terms of an automated redesign and process planning for IM. For example, in a subsequent step, the determined part segments can be compared with a product database using methods of shape matching (e.g. [Reuter et al., 2006](#)). This enables an automated assignment of suitable semi-finished parts to the derived part segments. Furthermore, Computer-Aided Process Planning often makes use of graph theory, e.g. to determine and evaluate possible manufacturing sequences (e.g. [Huang et al., 2012](#)). In addition, the automated segmentation process allows multiple parts to be decomposed and analyzed in a short time. This could, for example, support the design engineer with the definition of standard and variant parts in the context of design for variety.

3.1.5. Verification

To verify the segmentation algorithm, a set of various models from the ABC (A Big Cad) model dataset is taken into consideration. The ABC dataset is a collection of one million Computer-Aided Design (CAD) models for the research of geometric deep learning methods and their applications. Most of the models are mechanical parts with sharp edges and well-defined surfaces ([Koch et al., 2018](#)). Figure 5 shows a selection of the decomposition results for some typical mechanical parts of the shape benchmark. All these models are decomposed automatically into meaningful segments by the algorithm (requirement 1 and 2). In the context of IM, this means that all determined part segments should be monolithically manufacturable. All further requirements for the algorithm, which were defined in section 3, could be fulfilled for the shapes considered in the verification. Therefore, the

algorithm seems to be suitable for a broader range of mechanical parts with similar shapes to those decomposed in this paper.

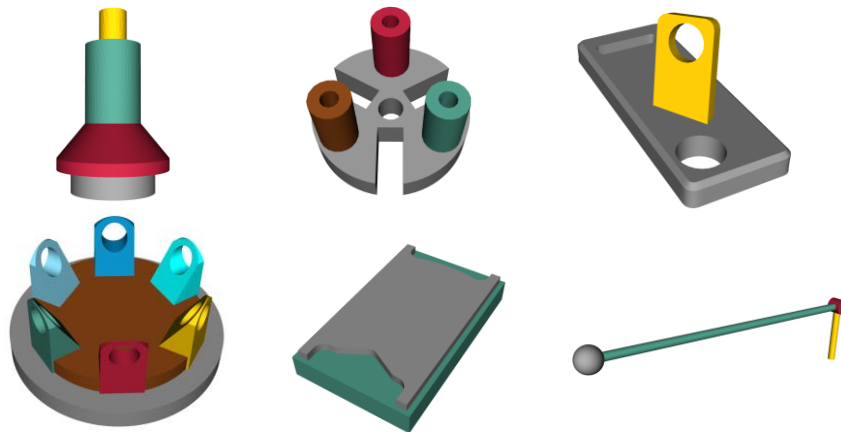


Figure 5. Exemplary results of the proposed mesh segmentation algorithm

4. Conclusion and outlook

This paper presented a novel approach for a part-based segmentation procedure, using a density-based clustering approach, which seems to be unique in the context of part-based mesh segmentation. For the algorithm, existing techniques of surface-based part segmentation have been reviewed, and important areas of activity are revealed. The approach contributes to a high degree of automation in the topic of surface-based part segmentation; thus, it keeps the required input parameters to a minimum. Furthermore, it faces the issue that many state-of-the-art segmentation algorithms require a determination of the number of segments a priori. The verification shows promising results and, due to the flexibility of the proposed approach, indicates a good transferability and extendability to more complex mechanical parts.

In the context of redesigning parts for IM, the algorithm allows an automated decomposition of a reference part into meaningful part segments, which can be manufactured monolithically. Moreover, it is discussed why the representation of the decomposed parts in a graph-based structure seems to be a promising foundation for further automation and analysis tasks regarding the design and process planning for IM.

In the future, the algorithm will be challenged and improved by the application to more complex parts and additional shape benchmarks. Moreover, research is being conducted into how the extracted graph can be automatically enriched with the functional view of a product, using theories of graph and shape matching along with further techniques of machine learning.

Acknowledgment

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 412330610. The authors hereby want to thank the financial support by the German Research Foundation (DFG DR 973/4-1, DFG VI 149/9-1).

References

- Agathos, A. et al. (2007), “3D Mesh Segmentation Methodologies for CAD applications”, *Computer-Aided Design and Applications*, Vol. 4 No. 6, pp. 827-841. <https://doi.org/10.1080/16864360.2007.10738515>
- Ankerst, M. et al. (1999), “OPTICS”, In: Davidson, S.B. and Faloutsos, C. (Eds.), *Proceedings of the 1999 ACM SIGMOD international conference on Management of data - SIGMOD '99*, Philadelphia, Pennsylvania, United States, 31.05.1999 - 03.06.1999, ACM Press, New York, New York, USA, pp. 49-60. <https://doi.org/10.1145/304182.304187>
- Aric, A. et al. (2008), “Exploring network structure, dynamics, and function using NetworkX”, *paper presented at Python in Science Conference (SciPy2008)*, August 2008, Pasadena, CA, USA.
- Arnold, P. and Rudolph, S. (2012), “Bridging The Gap Between Product Design And Product Manufacturing By Means Of Graph-Based Design Languages”, *paper presented at Proceedings of TMCE 2012*, May, 7 - 11, 2012, Karlsruhe, Germany.

- Dröder, K. et al. (2017), "Incremental Manufacturing: Design Aspects of flexible hybrid Manufacturing Cells for multi-scale Production", In: Schmitt, R. and Schuh, G. (Eds.), *7. WGP-Jahreskongress Aachen*, 5.-6. Oktober 2017, Apprimus Wissenschaftsverlag, Aachen, pp. 1-11.
- Dröder, K. et al. (2016), "Partial Additive Manufacturing: Experiments and Prospects with Regard to Large Series Production", *Procedia CIRP*, Vol. 55, pp. 122-127. <https://doi.org/10.1016/j.procir.2016.09.008>
- Gauthier, S. et al. (2019), "CAD-driven Pattern Recognition in Reverse Engineered Models", *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Prague, Czech Republic, 25.02.2019 - 27.02.2019, SCITEPRESS - Science and Technology Publications, pp. 244-254. <https://doi.org/10.5220/0007360702440254>
- Han, J., Kamber, M. and Pei, J. (2012), *Data mining: Concepts and techniques, The Morgan Kaufmann series in data management systems*, Elsevier/Morgan Kaufmann, Amsterdam.
- Hoffman, D.D. and Richards, W.A. (1984), "Parts of recognition", *Cognition*, Vol. 18 No. 1-3, pp. 65-96. [https://doi.org/10.1016/0010-0277\(84\)90022-2](https://doi.org/10.1016/0010-0277(84)90022-2)
- Huang, W., Hu, Y. and Cai, L. (2012), "An effective hybrid graph and genetic algorithm approach to process planning optimization for prismatic parts", *The International Journal of Advanced Manufacturing Technology*, Vol. 62 No. 9-12, pp. 1219-1232. <https://doi.org/10.1007/s00170-011-3870-9>
- Kalogerakis, E., Hertzmann, A. and Singh, K. (2010), "Learning 3D mesh segmentation and labeling", In: DeRose, T. (Ed.), *ACM SIGGRAPH 2010 papers*, Los Angeles, California, 7/26/2010 - 7/30/2010, ACM, New York, NY, pp. 1-2. <https://doi.org/10.1145/1833349.1778839>
- Koch, S. et al. (2018), *ABC: A Big CAD Model Dataset For Geometric Deep Learning*.
- Laga, H. et al. (2019), *3D Shape Analysis: Fundamentals, Theory and Application*, John Wiley & Sons, Inc. <https://doi.org/10.3726/978-3-653-05106-3/4>
- Liu, R. and Zhang, H. (2004), "Segmentation of 3D meshes through spectral clustering", *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings*, Seoul, Korea, 6-8 October 2004, IEEE, pp. 298-305. <https://doi.org/10.1109/PCCGA.2004.1348360>
- Pedregosa, F. et al. (2011), "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, No. 12, pp. 2828-2830.
- Reichler, A.-K. et al. (2019), "Incremental Manufacturing: Model-based part design and process planning for Hybrid Manufacturing of multi-material parts", *Procedia CIRP*, Vol. 79, pp. 107-112. <https://doi.org/10.1016/j.procir.2019.02.020>
- Reuter, M., Wolter, F.-E. and Peinecke, N. (2006), "Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids", *Computer-Aided Design*, Vol. 38 No. 4, pp. 342-366. <https://doi.org/10.1016/j.cad.2005.10.011>
- Richter, T., Inkermann, D. and Vietor, T. (2016), "Product Architecture Design As A Key Task Within Conceptual Design", *Proceedings of the DESIGN 2016 / 14th International Design Conference*, Dubrovnik, Croatia, May, 16 - 19, 2016, The Design Society, Glasgow, UK, pp. 1337-1346.
- Rodrigues, R.S.V., Morgado, J.F.M. and Gomes, A.J.P. (2018), "Part-Based Mesh Segmentation: A Survey", *Computer Graphics Forum*, Vol. 37 No. 6, pp. 235-274. <https://doi.org/10.1111/cgf.13323>
- Shamir, A. (2008), "A survey on Mesh Segmentation Techniques", *Computer Graphics Forum*, Vol. 27 No. 6, pp. 1539-1556. <https://doi.org/10.1111/j.1467-8659.2007.01103.x>
- Shlafman, S., Tal, A. and Katz, S. (2002), "Metamorphosis of Polyhedral Surfaces using Decomposition", *Computer Graphics Forum*, Vol. 21 No. 3, pp. 219-228. <https://doi.org/10.1111/1467-8659.00581>
- Theologou, P., Pratikakis, I. and Theoharis, T. (2015), "A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation", *Computer Vision and Image Understanding*, Vol. 135, pp. 49-82. <https://doi.org/10.1016/j.cviu.2014.12.008>
- Virtanen, P. et al. (2019), *SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python*.
- Zhang, J. et al. (2012), "Variational mesh decomposition", *ACM Transactions on Graphics*, Vol. 31 No. 3, pp. 1-14. <https://doi.org/10.1145/2167076.2167079>