

27th CIRP Design 2017

Analysis of Design Guidelines for Automated Order Acceptance in Additive Manufacturing

Jan-Peer Rudolph^{a,*}, Claus Emmelmann^{a,b}

^aLZN Laser Zentrum Nord GmbH, Am Schleusenengraben 14, 21029 Hamburg, Germany

^bInstitute of Laser and System Technologies, Hamburg University of Technology, Denickestraße 17, 21073 Hamburg, Germany

* Corresponding author. Tel.: +49-40-484010-735; fax: +49-40-484010-999. E-mail address: jan-peer.rudolph@lzn-hamburg.de

Abstract

Additive manufacturing (AM) is increasingly used in industrial production. Compared to conventional manufacturing technologies, such as milling and casting, AM offers a high degree of design freedom. Nevertheless, still some manufacturing restrictions and design guidelines have to be considered to ensure a flawless production. Therefore, a checking of design guidelines is a necessary step in order acceptance. Addressing this need, this paper presents an integrated analysis of design guidelines for an automated order acceptance.

In recent times, guideline catalogs for the design of additively manufactured parts have been developed. However, the analysis of a part's geometry with regard to these guidelines still requires a lot of manual work and expert knowledge. This paper introduces different algorithmic approaches, which automate the analysis and assessment of a part's geometry. Based on a preselection of guidelines from existing design catalogs for selective laser melting and sintering, this paper presents algorithms to automatically check the manufacturability of a part. The algorithms use the triangulated surface geometry (STL) of a part. They are implemented within a web-based platform for the automated order acceptance of additive manufactured parts. The evaluation compares the different algorithms regarding their efficiency and effectiveness.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference

Keywords: design guidelines, manufacturing restrictions, automation, order acceptance, design for manufacturing (DFM), additive manufacturing (AM), STL

1. Introduction and Background

Additive manufacturing (AM) technologies are increasingly used in the industrial production of plastic and metallic parts. This leads to the fact that more and more 3D printing service providers establish on the market. In recent times, the first service providers offer the option to place orders online [1]. AM builds up parts layer by layer based on given 3D geometry data [2,3]. Although AM offers a high degree of design freedom, still some manufacturing restrictions remain to ensure a flawless generation process [4,5].

In this paper, we present an approach for an integrated check of manufacturing restrictions and design guidelines as part of an automated order acceptance, with focus on selective laser melting (SLM) and selective laser sintering (SLS). Based on the results, parts can immediately be accepted and go into production (see Fig. 1) or rejected, if they cannot be produced.



Fig. 1. Integration of a web-based order acceptance into the AM process chain

The presented algorithms are prototypically implemented in a web-based platform, which simplifies the process of order acceptance. The customer can upload the geometry data of a part via an online form (see Fig. 2). Subsequently, the geometry is checked and critical areas are marked.

The paper is structured as follows: Section 2 presents the selected manufacturing restrictions and design guidelines, which are considered for an automated checking and are implemented within this paper. Section 3 describes the algorithms for checking a part's size. Section 4 introduces the algorithms for checking design guidelines for walls, gaps, cylinders, and boreholes. Section 5 evaluates and compares the different approaches. Section 6 gives a conclusion and outlook.

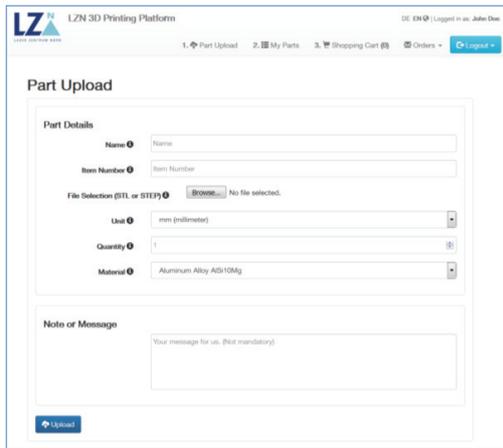


Fig. 2. Part upload via the implemented web platform

2. Preselection of Manufacturing Restrictions and Design Guidelines

In order to ensure a fault-free AM process and provide required qualities (e.g. shape and positional tolerances), certain manufacturing restrictions and design guidelines have to be considered. In recent research, process- and material-specific guideline catalogs have been developed [5–9].

To select these guidelines, which allow an automated checking based on a part's STL file, existing design catalogs are analyzed. Main selection criteria are the available and required input information for a design check. Thus, design guidelines, which are based on a fixed orientation of the part in the build chamber, are not taken into account, since the orientation is not known at the time of the online upload. Table 1 gives an overview of the selected restrictions and guidelines. The developed checking algorithms for these guidelines are presented in Section 3 and 4.

Restriction or guideline	Figure	Description
Part dimensions		
Part dimensions		The part has to fit into the build chamber of the generating machine. If necessary, the part must be oriented appropriately.
Walls and gaps		
Wall thicknesses		Wall thicknesses should not be below a certain limit to ensure a reliable and clean generation. The limit can depend on the orientation (angle to the build platform).
Gap dimensions		In order to avoid powder accumulations, merging of opposite areas within the part and facilitating removal of powder in the post-processing, gap dimensions should not fall below certain limits.
Cylinders and boreholes		
Cylinder diameters		The diameter of cylindrical structures should not be below a certain limit for a reliable and clean generation process.
Borehole diameters		In order to avoid powder adhesion, borehole diameters should not fall below a certain diameter.

Table 1. Overview of the selected manufacturing restrictions and design guidelines for an automated checking (example figures by [5])

The limits or thresholds, which decide whether a part can be produced or not, are implemented as configurable

parameters of the developed checking routines. Therefore, the algorithms are applicable on different materials and manufacturing processes. Exemplary for the materials polyamide PA12 (SLS) and titanium alloy TiAl6V4 (SLM), the limits, which can be found in the literature, are shown in Table 2.

Restriction or guideline	Polyamide (PA12)	Titanium (TiAl6V4)
Part dimensions		
Part dimensions	depends on build chamber of machine	depends on build chamber of machine
Walls and gaps		
Wall thicknesses	0.5 – 0.7 mm	0.3 – 0.4 mm
Gap widths	0.4 – 0.9 mm	0.2 mm
Cylinders and boreholes		
Cylinder diameters	1.4 mm	0.5 mm
Borehole diameters	1.2 – 1.4 mm	2 mm

Table 2. Limits for PA12 and TiAl6V4 (values taken from [5,6])

As already introduced, the algorithms expect the STL data of a part as an input. The STL format (Standard Triangulation Language) has established as a de facto industry standard in AM [10,11]. This is also the reason why the presented algorithms are based on the STL format. In order to execute the algorithms on CAD data (e.g. STEP), the input data of a part must be converted into STL previously. The algorithms are implemented in Java. A visualization, which shows the part and its critical areas, is written in JavaScript and WebGL.

3. Check of Part Size

The size of a part (or its dimensions) is a fundamental manufacturing restriction, which decides whether a part can be produced. In order to check whether a part fits spatially in a given build chamber of a generating machine, three approaches have been developed. All algorithms get the STL data of a part as an input. In the worst case the algorithms show a linear behavior, and thereby have a computational complexity of $O(n)$.

3.1. Complete Point Cloud

This algorithm uses the complete point cloud, which is given by the STL. This point cloud is systematically rotated until a suitable orientation (spatial limits of the build chamber are not exceeded) is found. The algorithm follows the sequence shown in Fig. 3. The preprocessing eliminates duplicate points, which can occur in STL files [11]. Due to the centering of the part and the build chamber, rotations in the range of 0° to 90° are sufficient. The accuracy of the solution depends on the configurable angular step size α .

The execution time of the algorithm mainly depends on the number of rotations to be performed as well as on the number of points to be checked per rotation. In the worst case all rotation combinations are traversed and per rotation all points are checked. The following two approaches (*bounding box* and *surrounding sphere*) try to reduce the execution time.

3.2. Bounding Box

Based on the basic algorithm, shown in Fig. 3, this approach reduces the number of tested points in the preprocessing. The aim is a reduction of the execution time. Instead of checking all points, the bounding box of the part is

used and only the eight vertices of the bounding box are rotated. Since the bounding box can also contain empty space, existing solutions might not be found. This means that acceptable solutions might be rejected. However, the algorithm can be used for a preselection of parts.

1. Center the part (based on its bounding box) and the build chamber in the origin (0,0,0).
2. Calculation or selection of the points to be checked (preprocessing).
3. Form a rotation combination $(\alpha_x, \alpha_y, \alpha_z)$ around the x-, y-, and z-axis with a fixed angular step width α :
 $\alpha_x = \alpha_x + \alpha$, while $0 \leq \alpha_x \leq \pi/2$
 $\alpha_y = \alpha_y + \alpha$, while $0 \leq \alpha_y \leq \pi/2$
 $\alpha_z = \alpha_z + \alpha$, while $0 \leq \alpha_z \leq \pi/2$
4. Execution of the rotation defined in (3.) and check whether all selected points lie in the build chamber. If a point is outside of the build chamber, go to (3.) and try another rotation. If all points lie inside the build chamber, a suitable orientation is found and the algorithm terminates.
5. Termination of the algorithm, because no suitable orientation is found.

Fig. 3. Basic algorithm for part size checking

3.3. Surrounding Sphere

While the first approach (*complete point cloud*) is computationally quite intensive, the second approach (*bounding box*) can lead to wrong results. The algorithm, described in this section, tries to overcome these disadvantages, to enable an efficient and effective size check. Similar to the second approach, the amount of points is limited by a preprocessing: only points, which lie outside of the minimal work area, are checked. The minimal work area can be described by a sphere, whose diameter has the length of the shortest side of the build chamber (see Fig. 4). Points outside of the sphere might also lie outside of the build chamber. These points are to be examined. The other points are ignored.

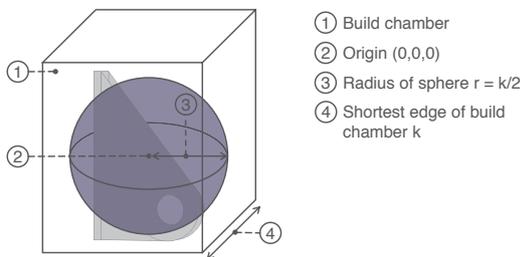


Fig. 4. Surrounding sphere, which describes the minimal work area

4. Check of Design Guidelines

This section presents the developed algorithms to check wall thicknesses, gap dimensions, cylinders and boreholes based on given STL data of a part. First, Section 4.1 presents how individual critical areas are detected. For the overall checking of an entire part two approaches are described: Section 4.2 presents a brute-force-based approach. Section 4.3 presents an advanced approach, which restricts the checked space to increase efficiency.

4.1. Detection of Critical Structures

First, we describe how critical walls and gaps are identified. Then, the identification of boreholes and cylindrical structures is presented.

4.1.1. Walls and Gaps

In order to measure wall thicknesses and gap dimensions, the distance between two opposite facets is determined. If the distance is smaller than the material-specific limits (see Table 2), the structure or area is classified as critical. Two facets are considered to be opposite, when the angle between their normal vectors is greater than a certain threshold value α_{opp} . This is illustrated by Fig. 5 for $\alpha_{opp} = 90^\circ$. In order to measure the distance between two triangles a method, described by Ericson, is used and implemented [12].

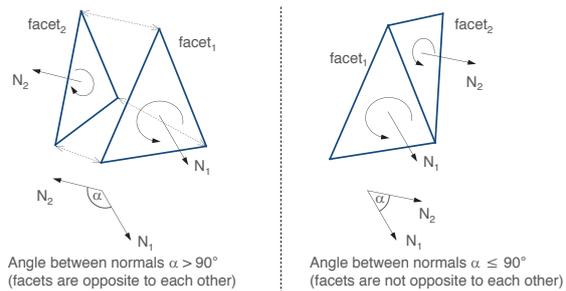


Fig. 5. Illustrating the definition of two opposing triangle facets

The STL format describes an object by its triangulated surface geometry. For each triangle the three vertices and its normal vector are stored. By definition, the normal vector of a triangle points to the outside of the described object [11]. Therefore, the distinction between wall thicknesses and gaps is based on the directions of the triangle normal vectors. If the normal vectors of two opposite facets point away from each other, it is a wall. If the normal vectors point towards each other, it is a gap (see Fig. 6).

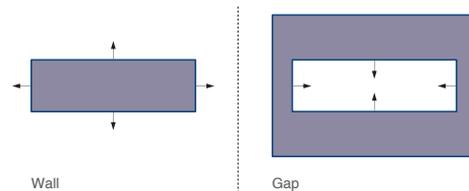


Fig. 6. Detection of walls and gaps

4.1.2. Cylinders and Boreholes

Since the limits for cylinders and boreholes differ considerably from those for wall thicknesses and gap dimensions (e.g. in case of titanium 2mm for boreholes compared to 0.2mm for gaps, see Table 2), cylinders and boreholes are treated separately.

For the detection of curved structures, the neighbors of a facet (surrounding triangles) are analyzed. Surrounding triangles of a facet are those which directly adjoin the facet and at least share one common point. Fig. 7 illustrates the three different cases: straight, convex, and concave structures. In case of convex structures the normal vectors diverge. In case of concave structures the normal vectors converge.

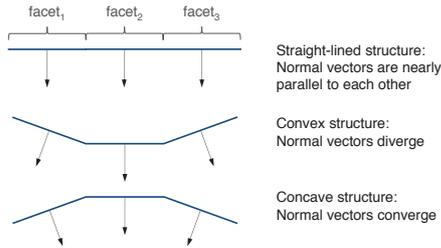


Fig. 7. Distinction between straight and curved structures

The distinction between convex and concave structures is used to detect cylinders and boreholes. A cylinder is present when the two facets, which are to be examined, lie in a convex environment and the normal vectors have opposite directions (~180°). This is illustrated in Fig. 8. The measured distance between the facets is then compared with the allowed cylinder diameter. Otherwise, the limit for the minimum wall thickness is applied. In case of a borehole, the facets, which are to be checked, are located in a concave environment.

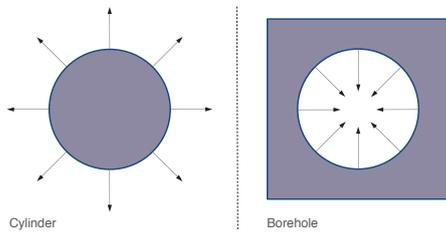


Fig. 8. Detection of cylinders and boreholes

4.2. Complete Space

For the overall checking of an entire part, the first method follows a brute-force approach. The position of each triangle to each other triangle is checked. In each individual check, the distance between the triangles and the relative positions of the triangles (angle and directions of the normal vectors) are determined. The computational complexity is $O(n^2)$, where n is the number of facets in the input file.

4.3. Subdivided Space

The calculation of the distances between each triangle to each other triangle is time-consuming and resource-intensive. Therefore, the following approach reduces the number of necessary checks. In a preprocessing the bounding box of a part is recursively subdivided into smaller, equal-sized subspaces (cuboids) (see Fig. 9). Each subspace stores the facets lying in it or adjacent to it. In order to check whether a facet intersects a subspace the method of Akenine-Möller is implemented [13].

The distance measurement is only performed for these triangles, which are assigned to one of the subspaces intersecting a sphere with a defined radius around the checked facet. Fig. 10 illustrates the principle in a simplified two-dimensional representation.

Compared to the brute-force approach (*complete space*) the method can significantly reduce the number of executed

checks. In the worst case (large radius of the sphere or rough subdivision of the bounding box) the computational complexity is $O(n^2)$.

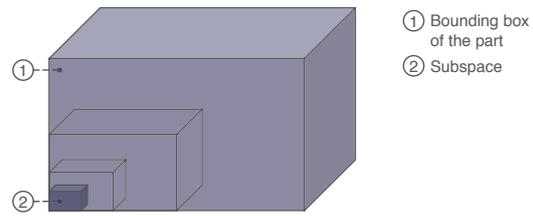


Fig. 9. Recursive subdivision of a part's bounding box (exemplary for one subspace)

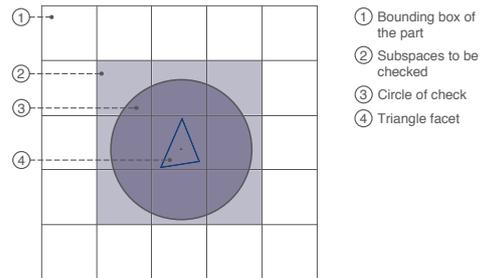


Fig. 10. Selection of the subspaces to be checked (two-dimensional)

5. Evaluation

The evaluation divides into two parts. First, the algorithms for the check of a part's size are investigated and compared. Afterwards, the two approaches for the design guidelines checking are evaluated.

The experiments were executed on following hard- and software: Intel Core i5-2430M (2.40 GHz) with 4 GB RAM and Windows 7 Professional (64 bit) with Java runtime environment JRE1.8.0_91.

5.1. Check of Part Size

The evaluation of the size check investigates the efficiency of the three presented algorithms (*complete point cloud*, *bounding box*, and *surrounding sphere*). Therefore four test scenarios are created, which are shown in Fig. 11 and tessellated with different accuracies. The angular step width for the rotation is set to $\alpha = 1^\circ$.

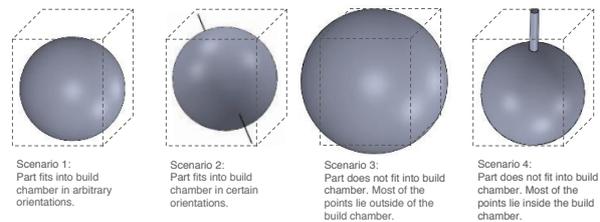


Fig. 11. Test scenarios for the evaluation of the size check

5.1.1. Scenario 1

In Scenario 1 the part fits into the build chamber in every orientation. Thus, no rotation of the part is necessary. The *complete point cloud* checks all points, the *bounding box* eight

points, and the *surrounding sphere* no points, since all points are excluded in the preprocessing.

As shown in Fig. 12, the *bounding box* is by far the fastest method (about 180 times faster than the *surrounding sphere*). The *complete point cloud* and the *surrounding sphere* behave roughly the same. The proportion of preprocessing is almost 100% for both approaches (see Fig. 14). Since only one orientation is tested, the number of points to be tested per rotation (see Fig. 13) is not the decisive factor of influence.

5.1.2. Scenario 2

In Scenario 2 the part fits into the build chamber, but not in any orientation. In average the *complete point cloud* and *surrounding sphere* need 3231 rotations until a suitable position is found. The *bounding box* does not find a suitable orientation.

Fig. 12 shows that for the *complete point cloud* the main driver for the execution time is no longer the preprocessing, but the average number of points tested per rotation (see also Fig. 13). The *surrounding sphere* is in average 4.7 times faster than the *complete point cloud*. Thus, the *surrounding sphere* is the approach to be preferred in this scenario.

5.1.3. Scenario 3

In Scenario 3 the part does not fit into the build chamber in any orientation, and most of the points lie outside. So, all 729000 rotations are traversed.

The approach of the *surrounding sphere* cannot show its advantages, since the preprocessing does not sort out any points. This is also shown by the execution times in Fig. 12. A closer look also shows that the preprocessing takes the majority of the total execution time (up to over 80%, see Fig. 14).

5.1.4. Scenario 4

In Scenario 4 the part does not fit in the build chamber, but most of the points lie inside of the build chamber.

The runtime behavior hardly differs from Scenario 3. For the *complete point cloud*, slightly more points are tested per rotation than in Scenario 3 (see Fig. 13), which has no significant effect on the execution time. However, as shown in Fig. 15, in the worst case (per rotation nearly all points have to be traversed) the *complete point cloud* behaves worse than the *surrounding sphere*, since it significantly reduces the amount of points in preprocessing.

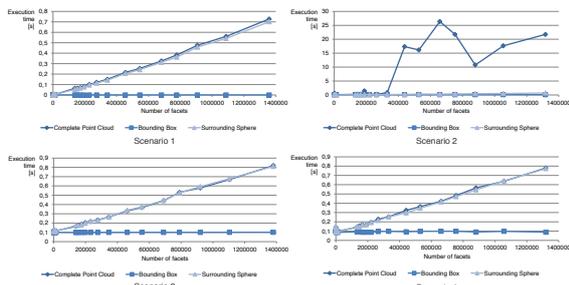


Fig. 12. Execution times of the size check

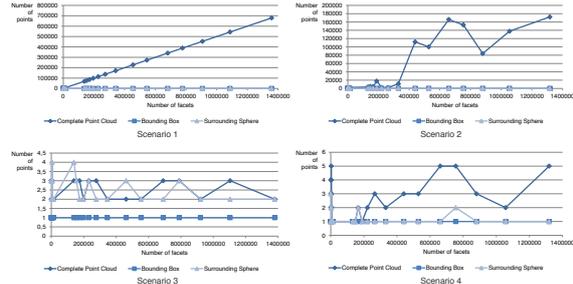


Fig. 13. Average number of points tested per rotation in the size check

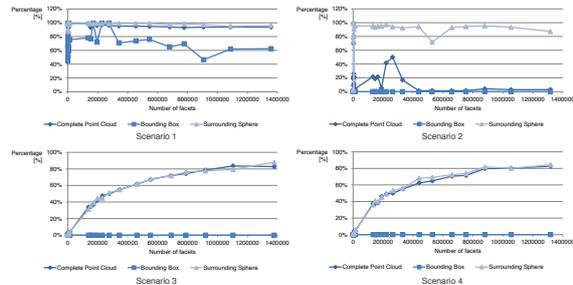


Fig. 14. Percentages of the preprocessing on the total execution time of the size check

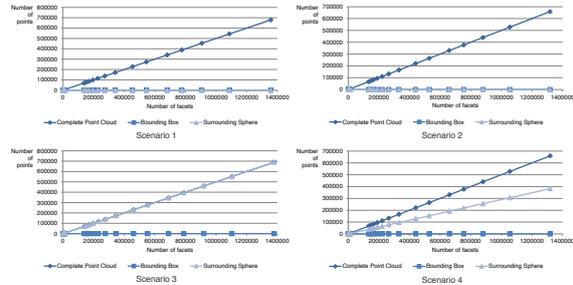


Fig. 15. Number of points to be checked per rotation in the worst case of the size check

5.1.5. Summary

In the experiments the algorithms behave (maximal) linear. This confirms the analyzed complexity $O(n)$. Moreover, the experiments show that the method of the *surrounding sphere* has significant runtime advantages over the *complete point cloud*. As a rule, this is the faster algorithm or in the worst case as fast as the *complete point cloud*.

Overall the *bounding box* is the fastest method, but can lead to false results (see Scenario 2). Parts which fit into the build chamber can mistakenly be classified as not suitable. Therefore, a single use of the *bounding box* as a test method is not useful. The *bounding box* should only be used for a preselection.

5.2. Check of Design Guidelines

Efficiency and effectiveness of the two presented algorithms for the check of design guidelines (*complete space*, *subdivided space*), are evaluated on three test specimens, which are usually used for the empirical determination of material-dependent manufacturing restrictions (see Fig. 16).

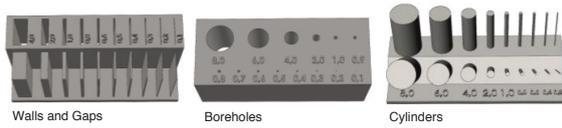


Fig. 16. Test specimens for evaluating the checking of wall thicknesses, gap dimensions, cylinder and borehole diameters (CAD data by [14])

5.2.1. Efficiency

In order to evaluate the efficiency, the three sample geometries are tessellated with different accuracies. The results of the execution time measurements are shown in Fig. 17. It becomes clear that the execution time behaves square-shaped to the number of facets in the input file for both algorithms. This confirms the analyzed complexity of $O(n^2)$. Thereby, the advanced approach of the *subdivided space* behaves much faster than the *complete space*. On average it behaves 2.3 times faster for walls and gaps, 1.7 times for boreholes, and 3 times for cylinders.

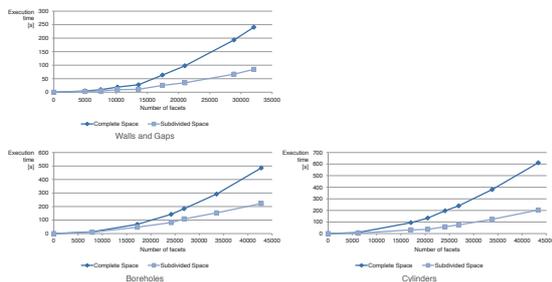


Fig. 17. Execution times of the design guidelines check

5.2.2. Effectiveness

For the evaluation of the effectiveness the three samples are used with the configuration for TiAl6V4 (see Table 2). An implemented visualization marks the critical facets in red color. For the test cases both approaches detect all critical areas and mark the same number of critical facets. The results are visualized in Fig. 18.

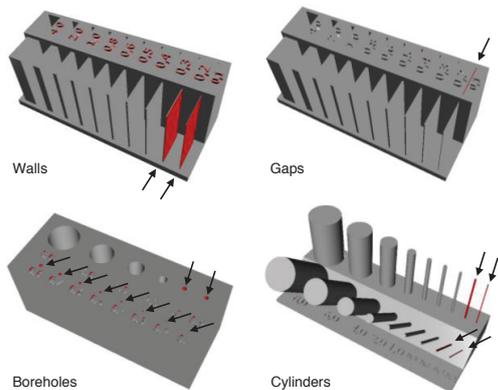


Fig. 18. Detected critical areas in the testing of TiAl6V4 on wall thicknesses, gap dimensions, boreholes, and cylinders

6. Conclusion and Outlook

This paper presents different algorithmic approaches for an automated checking of manufacturing restrictions and design guidelines for AM. In summary, the algorithm of the *surrounding sphere* should be used for the check of the part size, and the approach of the *subdivided space* should be used for the check of design guidelines (wall thicknesses, gap dimensions, boreholes, and cylinders). Through that, an efficient and effective order acceptance process can be implemented.

The presented checking algorithms do not consider the orientation of a part in the build chamber, since the final build orientation is not known at the time of the web-based order acceptance. Thus, a topic for future research is an accurate estimation of the build orientation and its integration into the design check. Based on the preselection of guidelines (Section 2), this paper focuses on algorithms for the checking of part sizes, walls, gaps, boreholes, and cylinders. The development of further algorithms for additional design guidelines is another topic for future work.

References

- [1] Wohlers Associates, Inc.. Wohlers Report 2016 - 3D Printing and Additive Manufacturing State of the Industry - Annual Worldwide Progress Report Fort Collins, CO, USA; 2016. ISBN 978-0-9913332-2-6.
- [2] Gebhardt, A.. Generative Fertigungsverfahren: Additive Manufacturing und 3D Drucken für Prototyping - Tooling - Produktion. München: Carl Hanser Verlag; 2013. ISBN 978-3-446-43651-0.
- [3] Herzog, D., Seyda, V., Wycisk, E., Emmelmann, C.. Additive manufacturing of metals. Acta Materialia 2016;117:371 – 392.
- [4] Emmelmann, C., Sander, P., Kranz, J., Wycisk, E.. Laser Additive Manufacturing and Bionics: Redefining Lightweight Design. Physics Procedia 2011;12, Part A:364–368. Lasers in Manufacturing 2011 – Proceedings of the Sixth International WLT Conference on Lasers in Manufacturing.
- [5] Kranz, J., Herzog, D., Emmelmann, C.. Design guidelines for laser additive manufacturing of lightweight structures in TiAl6V4. Journal of Laser Applications 2015;27(S1).
- [6] Wegner, A., Witt, G.. Konstruktionsregeln für das Laser-Sintern. Zeitschrift Kunststotechnik 2012;8(3):252–277.
- [7] Adam, G., Zimmer, D.. Design for Additive Manufacturing - Element transitions and aggregated structures. CIRP Journal of Manufacturing Science and Technology 2014;7(1):20 – 28.
- [8] Adam, G., Zimmer, D.. On design for additive manufacturing: Evaluating geometrical limitations. Rapid Prototyping Journal 2015;21(6):662–670.
- [9] VDI-Gesellschaft Produktion und Logistik, . VDI 3405 Blatt 3 – Additive Fertigungsverfahren - Konstruktionsempfehlungen für die Bauteilfertigung mit Laser-Sintern und Laser-Strahlschmelzen. VDI-Richtlinien: Verein Deutscher Ingenieure; Berlin: Beuth Verlag; 2015.
- [10] Danjou, S., Köhler, P.. Vorbereitung von CAD-Konstruktionsdaten für den RP-Einsatz - eine Schnittstellenproblematik. RTeJournal – Fachforum für Rapid Technologie 2008;2008(5).
- [11] Kai, C.C., Jacob, G.G.K., Mei, T.. Interface between CAD and Rapid Prototyping systems. Part I: A study of existing interfaces. The International Journal of Advanced Manufacturing Technology 1997;13(8):566–570.
- [12] Ericson, C.. Real-Time Collision Detection. Boca Raton, FL, USA: CRC Press, Inc.; 2004. ISBN 978-1-558-60732-3.
- [13] Akenine-Möller, T.. Fast 3d triangle-box overlap testing. In: ACM SIGGRAPH 2005 Courses. New York, NY, USA: ACM; 2005, p. 8.
- [14] LZN Laser Zentrum Nord GmbH. Sample geometries for the empirical determination of material-dependent SLM manufacturing restrictions. CAD data; 2016.