Received 27 January 2021; revised 19 February 2021; accepted 20 February 2021. Date of publication 23 February 2021; date of current version 17 March 2021. The review of this paper was arranged by Editor A. W. Colombo.

Digital Object Identifier 10.1109/OJIES.2021.3061610

Knowledge-Driven Manufacturability Analysis for Additive Manufacturing

MANUEL MAYERHOFER[1], WILFRIED LEPUSCHITZ [1] (Member, IEEE), TIMON HOEBERT[1], MUNIR MERDAN[1], MARTIN SCHWENTENWEIN [2], AND THOMAS I. STRASSER [3] (Senior Member, IEEE)

[1]Practical Robotics Institute Austria, 1210 Vienna, Austria
[2]Lithoz GmbH, 1060 Vienna, Austria
[3]Institute of Mechanics and Mechatronics, Faculty of Mechanical and Industrial Engineering, TU Wien, 1060 Vienna, Austria

CORRESPONDING AUTHOR: THOMAS I. STRASSER (e-mail: thomas.i.strasser@ieee.org).

This work was supported in part by the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology in the frame of the “Production of the Future” program under contract FFG 864798 and in part by the Vienna Business Agency in the frame of the “Research” program for the project ANALYTIC (proposal ID 2783347).

ABSTRACT Additive Manufacturing (AM) evolved recently from a rapid prototyping process to a standard manufacturing tool. Nevertheless, it is still not a widely used method due to different process-related challenges. In recent years printer technologies and possible printable materials emerged but there are still challenging demands on the printing process. Hence, it is of vital importance to inspect the manufacturability of the designed parts. This work focuses on the not yet widely researched ceramic printing with the Lithography-based Ceramic Manufacturing (LCM) processes. It presents a knowledge-driven framework able to automatically examine geometric properties of a part and compare it to AM guidelines. As a knowledge base, an ontology is used which contains information about the capabilities of AM processes, printers and materials. The manufacturability system uses triangle-based mesh processing algorithms to recognize features and check the guidelines necessary for LCM. The evaluation shows the feasibility of manufacturability analysis with the developed framework and its limitations.

INDEX TERMS Additive manufacturing, computer-aided design, knowledge representation, manufacturability analysis, ontology.

I. INTRODUCTION

AM, also referred to as 3D printing offers great potential in comparison to conventional manufacturing methods such as milling, casting or turning, due to the fast development of printer technologies and printable materials [1]. The advantages of AM technologies are the faster turnaround time and the feasibility of producing complex geometries with different materials [2]. Lately, the costs of 3D printers have fallen dramatically causing that an increasing number of people design individual object models as a hobby and make them available on platforms such as Thingiverse[1] [3]. However, often limitations arise in the form of minimum feature size producible by the process [4]. Moreover, not every Computer-Aided Design (CAD) model can also be produced with AM. Although AM offers a high degree of design freedom, still some manufacturing restrictions remain to ensure a faultless object creation. The manufacturability of an object, in reality, is dependent on various parameters, e.g. (i) AM process, (ii) printer resolution, (iii) layer thickness, (iv) the material used and so on [5]. Nevertheless, engineers and users of additive technologies often lack the awareness of manufacturing considerations leading to lower quality parts and print failures. Extensive knowledge and understanding of AM constraints and restrictions are needed for industrial adoption of the technology for end-use part production [6]. Users have to look for design guidelines, which provide information on how to design a specific, printable geometry in the literature or on 3D printing service platforms, for instance, 3D Hubs.[2] This process is time-consuming and it is also difficult to get a full overview of these diverse guidelines. Consequently, AM

[1]Thingiverse, available at https://www.thingiverse.com/

[2]3D-Hubs available at https://www.3dhubs.com/

service providers receive a huge number of unprintable object models and those then need to be checked manually by experts regarding their manufacturability [7].

In this context, a system is required that can support engineers through the design process of a specific model and assist by configuring process parameters [8]. These requirements imply a need for the development of a knowledge base able to capture all AM relevant information in a manner that can provide future sharing and integration. This work tackles these issues and introduces a knowledge-based framework that considers specific restrictions of the AM process and automatically analyzes the manufacturability of a particular object. Modeling a comprehensive system for all AM processes is difficult because the requirements for each process differ slightly. Consequently, this work focuses on one type of process as a starting point, which is LCM.

This work contributes to the improvement of the automated manufacturability analysis in the domain of AM by two parts: The first part, which leads to an improvement of the automation process, is the development of an ontology about the LCM process and its manufacturability analysis. This ontology is intended to represent the most important design guidelines for the various technologies as well as material dependencies. However, there is an overlap of the most common LCM guidelines with the generic AM process so that the system can be easily extended to other domains. The second part consists of a software system that checks the manufacturability of the object models, enabled by the interconnection with the ontology for information transfer. The contribution of this work is broadened by the description of all implemented geometry analysis algorithms, with an added algorithm runtime analysis. Additionally, the authors contribute a novel algorithm for overhang classification, as well as a novel integration of a kd-tree data structure for wall thickness analysis algorithms to enable a more efficient runtime of this existing algorithm. This analysis system should help AM service providers to reduce the number of non-printable object models by performing a pre-process manufacturability check. With this software package, the engineers can evaluate their object models and get visual and textual feedback about the manufacturability with the specific AM process and material.

The basic idea of the proposed analysis system was already outlined in a previous work-in-progress paper [9]. It briefly described the general concept and the most important components of the proposed framework. In the meantime, the algorithms, the corresponding interfaces as well as the ontology have been significantly improved and tested, leading to concrete results, which are presented in this work. Therefore, this article provides an extensive discussion and it is structured in the following way: First, the process of lithography-based AM is introduced in Section II followed by the analysis of the related works and the state-of-the-art in the targeted domain in Section III. The framework architecture as well as an ontology representing the target system are presented in Section IV-A. Section V provides in detail the implemented framework whereas the evaluation of the approach is discussed in Section VI. Finally, Section VII concludes the work with an outlook on future research.

II. LITHOGRAPHY-BASED ADDITIVE MANUFACTURING

Ceramic is widely deployed in a large number of applications due to exceptional advantages related to corrosion resistance, the ability to tolerate high temperatures as well as for having excellent mechanical properties, such as hardness and stiffness [10]. However, a broader usage of this material for small series of products with complex geometries was so far limited due to high costs, extended production times as well as design restrictions of conventional production processes [11].

Recent advances in lithography-based AM technology facilitated the ceramic-based fabrication of high-performance ceramic products in small series or even as individual products. Lithography is based on the concept of photopolymerization, where a photosensitive suspension is cured layer-by-layer through exposure to light from the bottom side of the vat that is containing suspension. The suspension consists of solid ceramic particles well dispersed in a liquid photopolymer that serves as a binder between the ceramic particles and enables the precise shaping of the part. The build process occurs bottom-up after the building platform is moved downwards into the vat and the layer is cured through irradiation by a light engine, having then the building platform moved upwards to enable the recoating of the vat. These steps are repeated until the fabrication process is completed and the part, called a green body, is achieved. Afterwards, the green parts have to undergo additional thermal post-processing, i.e. debinding and sintering to convert them into dense ceramic parts. During the debinding step, the polymer network is burned off at temperatures up to 500 °C. The subsequent sintering in a high-temperature furnace enables densification by fusing the ceramic particles [10]–[12].

The technology allows the manufacturing of components with complex features, which cannot be fabricated with any other technologies. However, despite its clear advantages, extensive knowledge and understanding of AM constraints and restrictions are required for wider industrial adoption of LCM technology due to its novelty. AM offers high design freedom, but cannot print every model due to process, material and machine parameter restrictions. Especially for AM of ceramic components using the LCM technology, different process-specific challenges, such as minimum printable feature size, necessary support material, as well as the thermal post-processing that involves debinding and sintering accompanied by shrinkage of the printed component, need to be considered. In this context, most current guidelines check only for minimal feature size but do not distinguish between walls and pins or notches and holes. Moreover, a significant difference of the LCM process to other AM processes is the requirement of the curvature. The LCM process cannot print right angles and generally radii of curvatures below a certain radius are not possible, as they can result in cracks during thermal post-processing.

III. RELATED WORK

Manufacturability analysis has been researched through various aspects and significant efforts have been made to verify it [5]. However, geometry requirements are different from those of conventional manufacturing processes and the necessary information is not yet available in an applicable context [13].

A. DESIGN GUIDELINES

Design guidelines are created to make efficient use of AM and they support the design for production to enable robust manufacturing to ensure accuracy and function of the printed object. Restrictions are determined by experiments and documented in various forms. In recent years this has resulted in a large number of process-specific guidelines with different qualitative and quantitative information, which can be categorized into four levels [14]:

1) Endangered manufacturability/functionality of the object,
2) Loss in component quality,
3) Determination by the user without influence on manufacturability or component quality, and
4) Already taken into account in other guidelines.

The AM manufacturability guidelines also depend on the material and intended use. Each material has different properties and therefore has different printing possibilities. Kranz *et al.* [15] investigated in a study about design guidelines for laser AM of lightweight structures with the material Ti6AI4V. Meisel *et al.* [16] investigated quantitative design thresholds for PolyJet material jetting and identified several key manufacturing constraints: (i) Minimum feature size, (ii) minimum self-supporting angle and support material removal as well as (iii) special material jetting constraint.

Design guidelines are carried out by experimenting with specific machines and corresponding parameter settings with different materials. Generally, guidelines are examined depending on the process, as shown above for Laser AM [15] or Material Beam Method [16]. However, there are also approaches to consider design guidelines more globally and to investigate commonalities. The research project Direct Manufacturing Design Rules (DMDR)[3] has developed guidelines for AM for users in science, industry, and education. They developed a process-independent method for the development of design guidelines. The methodical approach made it possible to identify commonalities between the procedures and to derive guidelines from these that are valid for the considered AM methods. The guidelines were summarized in a design guideline catalogue [13], [17]. The developed guidelines are only valid for the boundary conditions considered in the DMDR project. However, guidelines are mostly dependent on the material and parameter settings of the machine. Therefore a follow-up research project Direct Manufacturing Design Rules 2.0 (DMDR 2.0) was started to extend the scope of the previously developed guidelines. Usually, the guideline description is only textual (compare [15], [16]) and is slightly different from unrelated sources. This leads to ambiguities and makes it difficult to have a consistent vocabulary for AM design solutions.

Jee *et al.* [18] propose a formal design guideline representation to standardize guidelines modularly. One example guideline is defined by Jee *et al.* [18] to demonstrate the structure: "Overhangs (angular) if designed at greater than 45 degrees of undercut angle and built by a metal-based powder bed fusion process, are self-supporting." Another approach to efficiently represent design guidelines is proposed by Mani *et al.* [19]. They proposed the Guide-to-Principle-to-Rule (GPR) approach. Design Rules (DR) are derived from Design Principles (DP) founded in Design Fundamentals (DF), which are abstracted from Design Guidelines. DF can be categorized into geometry related parameters, material parameters, and machine parameters related to Ullman *et al.* [20]. By definition of Mani *et al.* [19], DPs are logical correlations capturing process parameter and control parameters and DRs are specific correlations that provide needed insight into manufacturability. The concept makes it possible to encode new knowledge formally, consistently and expandable. Regarding the LCM procedure Scheithauer *et al.* [21] have described the opportunities and limitations in their work. Furthermore, two methods [18], [19] were presented to encode the knowledge formally, consistent and AM process independent and dependent. These methods are based on a guideline set with mathematical functions.

To summarize, the state of the art analysis showed that there is a lot of research in the area of design guidelines. Most approaches tackle specific AM technologies and their constraints, but there are also similarities as shown by Adam *et al.* [13]. Besides, there are several points of view on the manufacturability checks that are ultimately the same as for instance general feature constraints, such as minimum feature size, and constraints regarding feature categories, for example walls, gaps, holes, and pins, which are to be considered in every technology [22]. However, there are also specific geometry requirements for each process (see material jetting [16]) as well as requirements that are not necessary to be taken into account using certain technologies such as overhangs in the cases of binder jetting and selective laser sintering. In recent years, this has resulted in a large number of process-specific guidelines with different qualitative and quantitative information. Consequently, it is time-consuming and difficult to get an overview of the right guidelines needed for a specific purpose.

In order to achieve an automated manufacturability checking framework, the definition of guidelines is the first step that is necessary. As a second step, this information must also be encoded in a machine-readable manner. The deficiency of (structured) knowledge on Design for Additive Manufacturing (DFAM) has been recognized as one of the obstacles that hinders wider acceptance of AM in industry [23]. A way to formally define knowledge is to employ ontologies, which is discussed below.

[3]DMRC available at https://dmrc.uni-paderborn.de/de/inhalt/forschung/interne-projekte/direct-manufacturing-design-rules-20

B. ADDITIVE MANUFACTURING ONTOLOGIES

With the development of the semantic web, a possibility emerged that allows representing the existing knowledge in a systematic way for making information context searchable and functional for its intended use [24]. The use of ontologies makes such information models both human-readable and machine computable. Furthermore, ontologies can be shared and reused without the loss of computability [25], [26]. The current state of the art is well described by Kim *et al.* [27]. Moreover, they present an ontology providing an updated DFAM knowledge based on their previous work [28] also including the research from Dinar *et al.* [8]. Their work aims to formulate the ontology as generic as possible in the field of AM and incorporates on the one hand the GPR methodology [19], which is used for principles of design guidelines in AM (American Society for Testing Materials (ASTM) WK54586), and on the other hand the module concept by Jee *et al.* [18]. The final ontology resulted in a structure that has five top-level classes [27]: (i) Feature, (ii) Parameter, (iii) AM_Capability, (iv) AM_Process, (v) and Machine. Furthermore, Hagedorn *et al.* [29] proposed an ontology for Innovative Capabilities of Additive Manufacturing (ICAM). The work has the aim to facilitate innovative use of AM by linking knowledge from business with knowledge bases about the capabilities of AM processes and machines. Besides, Ali *et al.* presented the Additive Manufacturing Ontology (AMO), which is a common ontology designed to represent the AM Product Life Cycle and solve data integration challenges [30]. Liang proposed an ontology-oriented knowledge methodology for the generation of a novel knowledge in the AM production design process based on a novel ontology (AM-OntoProc) and the informal model [31]. Ko *et al.* presented an approach for automated and autonomous knowledge-driven AM design rule construction, that combines AM data, machine learning, and an ontology with knowledge graphs [32]. There are also some other approaches, which employ ontologies in the field of AM [26], [33], but not yet in combination with software that automatically checks production guidelines.

C. MANUFACTURABILITY MESH PROCESSING

To determine the manufacturability, the geometry of the object to be printed must be examined with feature recognition methods. These methods interpret the solid model in terms of predefined features, for example holes, pins, walls, gaps, and overhangs. The manufacturability analysis can be performed in two different ways depending on the view of the object in two or three dimensions.

1) TWO DIMENSIONAL OBJECT MODEL ANALYSIS

AM produces the solid model in a layer-wise process and each distinct layer should be printed without errors. Therefore, one possibility is to analyze the manufacturability in two dimensions at each layer. In geometry, the medial axis of an area is a set of points located in a kind of a geometric center of the area. The medial axis can be used to calculate distances and radii with the diameters of maximal circles [34]. Shape Diameter Functions (SDF) are widely used for the estimates of the local thickness of shapes. Jaiswal and Rai [4] use a variation of SDF for segmenting thin regions in slices. The SDF function maps the local diameter to each point on the boundary of the sliced object. Also, the interior angle at vertices can be used to determine sharp corners. By checking the internal angle between two edges, not printable corners can be recognized. Moreover, morphological operations can be used to identify small holes and intrusions as well as to identify small features, especially walls. Nelaturi *et al.* [2] use the morphological opening operation to form a printable map of the object. Jaiswal and Rai [4] use the morphological closing to detect small holes and intrusions in their manufacturability framework.

2) THREE DIMENSIONAL OBJECT MODEL ANALYSIS

When analyzing the object meshes in three-dimensional space, the mesh is considered as a volumetric mesh. The analysis can be performed with voxels, polygon meshes or heat kernel signature. The voxel-based representation uses voxels (cubes aligned to the cartesian coordinate system) to represent the shape of the object. Leary *et al.* proposed a voxel-based Cellular automata approach for the generation of AM support structures cite [35]. Tedia and Williams [7] developed a framework for performing a manufacturability analysis of parts to be manufactured using a voxel-based representation schema. The framework provides feedback on unfeasible features, minimum feature size, support material, orientation and manufacturing time for different build orientations. However, this approach does not categorize features into categories namely, walls, holes or cylinders. Thus, for example, no explicit query can be made for hole radii. On the other hand, a polygon mesh is a collection of vertices, edges, and faces that defines the shape of a polyhedral object. The faces consist of triangles, quadrilaterals or other simple convex polygons. Rudolph and Emmelmann [36] use the triangulated surface geometry for automatic analysis and assessment of a part's geometry. They check the common guidelines include the part size, wall thickness, gap size, hole size, and cylinder diameter. However, this approach is rather focused on relatively simple part geometries. For numerically generated structures, further guidelines have to be considered, such as those that describe the minimum and maximum cross-sectional areas and overhang without support structures.

Besides, during the calculation of some guidelines, for example, the overhang, the orientation of the product and the build direction needs to be taken into account. Allaire *et al.* used triangular 3D meshes for to find the optimal build orientation of the object to be manufactured as well as for the shape and topology optimization of supports. [37]. Allaire *et al.* used triangular 3D meshes for to find the optimal build orientation of the object to be manufactured as well as for the shape and topology optimization of supports [38]. A data-driven predictive model able to predict the printability of a given artefact is proposed by Mycroft1 *et al.* [39]. Tominski *et al.* [22] proposed a software-based design check for AM concept. He builds on the results of Rudolph and Emmelmann [36] and extends their solutions with suggestions for missing guidelines. Tominski *et al.* [22] also described the usage of the medial axis to calculate distances and radii. Finally, the Heat Kernel Signature (HKS) is a pointwise shape descriptor developed in computer vision [40]. HKS defines for each point in a shape a feature vector and is based on the concept of heat diffusion over a surface. HKS estimates the heat loss over time and the rate indicates the topological and geometric properties of a point on a given domain. Shi *et al.* [41] define the incremental value of an interval where the heat value on a node persists above a preset threshold as heat persistence value. The value is the area below the heat curve and can be computed as the integral of the heat function. However, only the features can be detected with this method and lengths, angles, as well as orientations, are required for the manufacturability analysis. These are calculated with the properties of the Singular Value Decomposition (SVD). The SVD gives the vertex distribution in the feature. The SVD indicates the main direction of the distribution, i.e. the propagation of the object in one direction and also the smallest distribution, the minimum propagation. With the HKS and SVD the approach by Shi *et al.* [41] supports the following guideline checks: minimum feature size, overhangs, surface orientation for self-supporting structure, minimum space between features and Maximum vertical aspect ratio.

In summary, guidelines are defined very specifically in the literature and are not presented in a uniform way, which would simplify the application for the manufacturability check. Furthermore, a new simple guideline structure should be defined, which makes it possible to assemble guidelines in a modular way. As can be seen in the guidelines from Kranz *et al.* [15], some guidelines are still given in qualitative information that is not machine-readable. This is also the case with the LCM process, where the current guidelines are available in the form of datasheets of materials or simple text form. The comprehensive work of Kim *et al.* [27] is a very good starting point for an ontology for an automatic manufacturability test. However, to be able to construct a query for the manufacturability the information structure of the DFAM ontology must be understood.

Besides, the geometry must be available in its defined format to carry out the test and the software is required to extract the necessary features. Besides, some of the analysis of the current implementation only includes general guidelines and do not distinguish between details. For example, most guidelines check only for minimal feature size but do not distinguish between walls and pins or notches and holes. Also, overhangs are not well addressed in the existing approaches, Moreover, some specifics of the LCM production method such as inability to print right angles, as well as radii of curvature below a certain radius, has not been addressed in any of the previous approaches.

D. OPEN ISSUES AND SHORTCOMINGS

To summarize, several open and unsolved issues exist in regard to automated guideline analysis for the LCM-process. The most relevant topics for the presented work are:

- Ontologies are missing for formalizing the LCM process and its manufacturability analysis guidelines.
- No software system incorporates an ontology of manufacturability guidelines to check the manufacturability of object models.
- Geometry analysis algorithms exist but they are not described in depth and miss implementation-specific details. Especially the run-times of these analysis algorithms are not analyzed and quantified in contrast to the input geometry size.
- The currently known wall thickness analysis algorithms are not run-time optimized and have the deficit of an exponentially growing run-time in contrast to the input geometry size.
- Currently, there are no geometry analysis algorithms that classify the different types of overhangs, which, for example, can be "T" or "H"-shaped.

In the following sections, an approach to address those issues is introduced and discussed in detail.

IV. AM MANUFACTURABILITY ANALYSIS SYSTEM

This section describes the architecture of the framework for manufacturability analysis. The proposed approach has to be able to automatically examine a given geometry, compare it to AM guidelines and give the user visualized information about the part manufacturability. The framework integrates a Mesh Analyzer and an Ontology with a cloud platform that incorporates an ordering form as well as a mesh visualizer (see Fig. 1).

A. MESH ANALYZER

The quality of the printed part depends on different aspects such as chosen build orientation, material properties, surface finish, enclosed voids and printer parameters. In this context, it is very important to correctly define all relevant parameters, e.g. the minimal size of walls and holes in the part. Thin walls could lead to thermal dissipation and cause different defects, such as un-melted powder inclusions, internal voids, cracks and shape irregularities [41]. Generally, the design process should be done in several loops considering the final function of the specific part but also performing a manufacturability analysis involving all relevant aspects. In this work, the manufacturability check is performed with the mesh analyzer. The analysis mainly operates in three-dimensional space with an exception in the cross-sectional area, where a layer (slice) is examined. The mesh analyzer uses the triangle representation of the objects to perform a triangle-based analysis. This work is based on the approach presented by Rudolph and Emmelmann [36], which uses the triangulated surface geometry in Standard Triangulation/Tesselation Language (STL) format

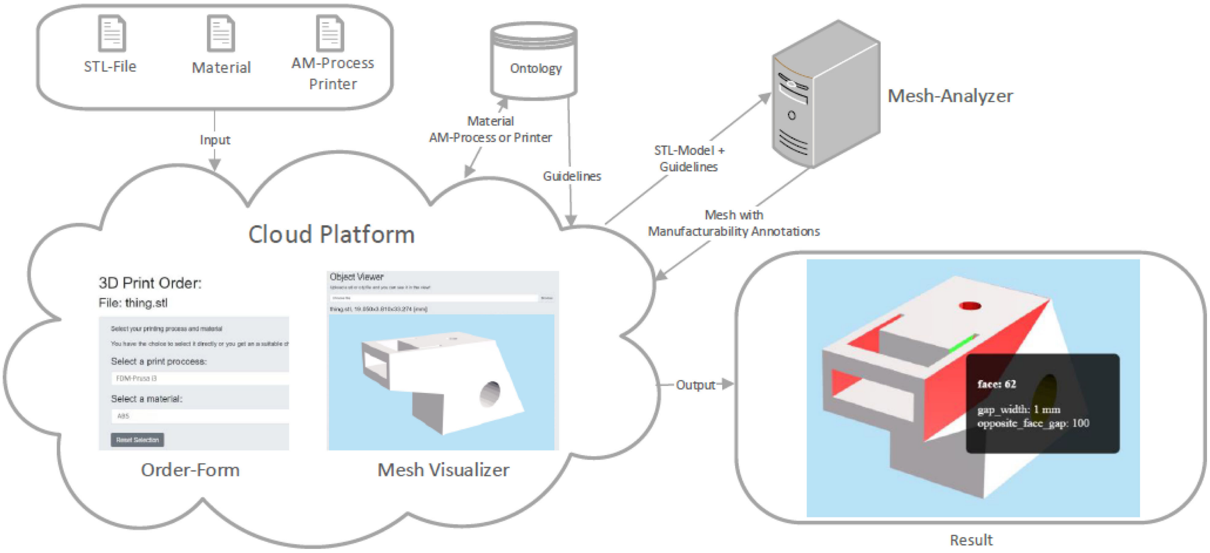


FIG. 1. Architecture of the framework for manufacturability analysis [9].

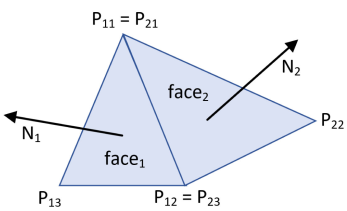


FIG. 2. Triangle connection with corner points and surface perpendicular in the STL format [9].

for automatic analysis and assessment of a part's geometry. The format contains three vertex points and a perpendicular for each triangle of the model (see Fig. 2).

By linking the triangles, the necessary geometric information, such as distances, radii, curvatures, and area sizes, can be calculated to verify the manufacturability features. However, as presented in Section III, Rudolph and Emmelmann's [36] approach is rather focused on relatively simple part geometries and only implements four common guidelines: wall thickness, gap width, hole radius, and pin radius. Besides, during the calculation of some guidelines, for example an overhang, the orientation of the product and the build direction needs to be taken into account. In this context, Tominski *et al.* [22] presented ideas on how to consider guidelines that take the building direction into account and use the normal vector to check the surface orientation as well as overhang angles. Furthermore, they introduced the idea to calculate the cross-section area with Gauss's area formula on one slice of the object.

Based on this background, the feature extraction algorithms were developed and adapted to the requirements of the LCM process. Within this context, the following guidelines for the LCM process have been examined: (i) wall thickness, (ii) gap width, (iii) hole radius, (iv) channel size, (v) pin radius, (vi) pin ratio, (vii) overhang without support structure (H, T-type) as well as (viii) (curvature) inner and outer radius. Besides, other algorithms for guideline checks that have not been investigated so far, such as the curvature or differentiation of the overhang type, are also developed. The related guidelines and applied algorithms are presented and discussed in the following subsections.

1) ALGORITHM RUNTIME – K-DIMENSIONAL TREE

The runtime of the algorithms depends on the number of triangles in the object. Due to the representation of the object as a triangle mesh, it is difficult to create curves. Many small triangles are used to create a more detailed curve, which can cause a very high number of triangles of an object. Depending on the triangulation resolution, the number of 1000 triangles is sufficient for small structures with few curves but it can be even half a million triangles for more complex structures. This leads to an immensely high runtime for algorithms that compare each triangle with all others.

The use of the space-partitioning data structure k-dimensional tree (k-d tree) [42] is a possible solution to the problem by grouping neighboring triangles for efficient search. Such a tree is a binary one that at each level of it divides the space along an axis with a hyperplane into two parts. The subdivision is performed according to the X, Y, Z axis, as depicted in Fig. 3. This kind of data structure has been used for computer graphics application for decades. The construction of a k-d tree, which is filled with triangles, works as follows: The root node of the tree forms the bounding box of the object. Each level of the tree subdivides the space into two parts. The two new nodes then form half of the bounding box of the previous level. The node to which the triangle is assigned is then evaluated. This is done by an intersection test with the enlarged triangle bounding box and the two other bounding boxes. The bounding box of the triangle is increased by the distance of the minimum wall thickness. It has to be enlarged because a triangle can be located at the edge of the bounding

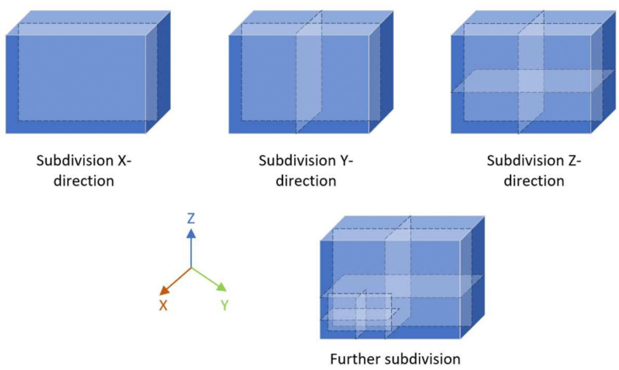


FIG. 3. kd-tree construction by subdivision.

box of one node and so no test with the other node would need to be done. If the intersection test is successful the triangle will be added to the node. This step is repeated recursively up to a termination condition. The applied termination condition therefore is the minimum wall thickness representing the leaf of the tree and the smallest bounding box.

For the application of the algorithms under usage of the k-d tree data structure, the tree must be traversed. In the case of the wall thickness algorithms, the leaf nodes containing the desired triangle are searched. Due to the tree structure, the algorithm only has a runtime of $O(n * log(n))$ [43], with n being the number of triangles. In the leaf node, the triangle is then only examined with the remaining triangles within the leaf node.

2) WALL THICKNESS AND GAP WIDTH

An important requirement for the printing process is the minimum wall thickness or, in the case of ceramic printing using the LCM process, also the maximum wall thickness (i.e., the distance between two surfaces). If the space between the two surfaces is filled with material, it is denoted as a wall. If there is no material between the two faces, it considered as a gap. The naive algorithm checks each perpendicular with each other perpendicular. Our implementation with the k-d tree only has to check a subset of neighboring triangles which are found with the tree data structure. Faces with an obtuse angle to each other (angle $>$ 90 degrees, dot product < 0) are considered opposite. Afterwards, it is checked whether the two perpendiculars stand to each other or opposite each other. The check is done by calculating the distance between the two triangles. The perpendicular of the respective triangle is added to the triangle vertices. First, the distance between the two vertices is calculated. Then the distance between the vertices and the distance between the vertices with the added perpendicular is calculated. The difference in length is used to determine whether they point towards each other or away from each other. If the second distance is larger than the first, it is a wall and the other way around a gap. Finally, the minimum distance between the two triangles is calculated using the algorithms by Ericson [44].

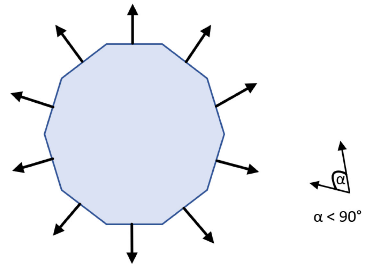


FIG. 4. Triangulated circle with perpendiculars.

3) HOLE RADIUS AND PIN RADIUS

Hole radii and pin radii are also standard guidelines in AM. Different requirements apply compared to wall thicknesses or gap width. Looking at the surface of a pin and a hole they are identical objects with different perpendiculars. Therefore, circular structures are located, which are then differentiated according to the perpendicular. Regarding pins, the perpendiculars point away from each other while concerning holes they point towards each other. At first, the algorithm searches for a contiguous structure that forms a circle (see Fig. 4). Here the perpendiculars are taken into account. Neighboring faces should have an acute angle (angle $<$ 90 degrees, dot product $>$ 0) to each other. Due to the construction of a pin using a CAD program to create a triangle mesh, two triangles are connected with the same perpendicular. Therefore, adjacent faces with the same surface perpendicular (angle $= 0\,^{\circ}$, dot product $= 0$) are included as well. The acute angle between the triangles is buffered and compared with the next possible adjacent triangles. If an adjacent triangle has the same angle or if the angle is zero, the triangle is included in the circular structure. This avoids finding other structures that do not form a circle. Two faces within the circular structure are determined which are at an angle of 180 degrees to each other. The minimum distance between these two triangles is calculated using algorithms by Ericson [44]. The distinction between pin and hole is analog to the distinction between walls and gaps as mentioned in the previous subsection.

Another guideline in LCM is the verification of the channel size. A channel is a continuous hole. This means that by looking through a straight channel, it is possible to see through the object. The check is done by analyzing the holes. From the extracted hole structure, two adjacent triangle faces are located at each hole. The perpendiculars are then compared to each other. If the two faces point in the same direction, it is a hole, if they point away from each other, it is a channel (see Fig. 5).

4) OVERHANG AND SURFACE ORIENTATION

Overhangs are structures that are at a special angle to the building direction. This means that the angle between the surface of the overhang structure and the building direction determines whether it is indeed an overhang structure or not. It is important to recognize overhang structures because they need a support structure from a certain length. If an object

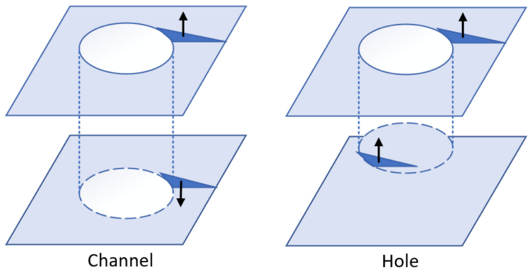


FIG. 5. Difference between a channel and a hole with adjacent perpendiculars.

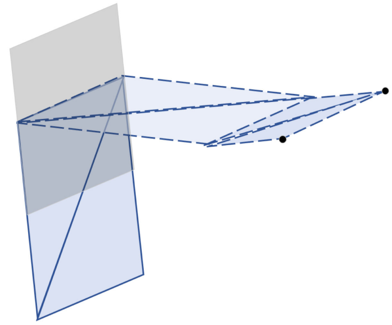


FIG. 6. Overhang length calculation with the maximal (concerning two points) minimal distance between plane (gray) and the two points (black).

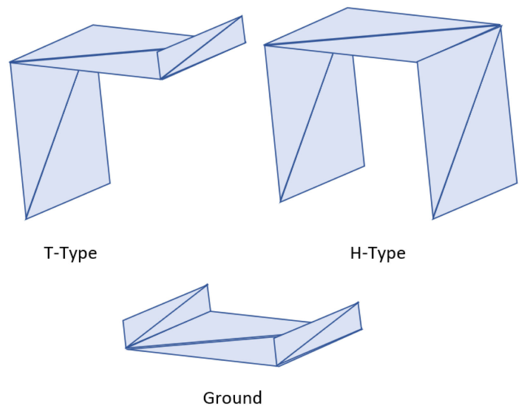


FIG. 7. Overhang types.

needs support structures, these must also be removed at a later process step and the removal of the support leads to reduced surface quality. Usually, an overhang requires a support structure from an angle of 25 degrees but this also depends on its length. There are also two distinctions, whether the overhang structure is unilaterally attached or multilaterally attached. A unilaterally attached overhang structure means that one end of the structure is not attached to a further part of the object. In the technical language it is denoted as T-type, because of the resulting T shape. Bilaterally attached overhangs are connected on both sides to other parts of the object and form a bridge. Consequently, this form is also denoted as H-type overhang in technical language. These two forms have different length constraints in regard to being printed and further processed without a support structure. The possible length of bilaterally attached overhangs without support structures is longer due to the greater stability. In the analysis, all faces are examined concerning the angle of the building direction. Faces that have a smaller angle than the critical angle will be further investigated, with the critical angle being the minimum angle necessary to print the segment without a support structure. Critical faces are then merged into connected components. Beginning and end of the structure are chosen in each connected component by choosing the two faces of the connected component with the largest distance. The length between the two faces and the overhang type is calculated thereafter.

At first, an attached face is searched. If there is a T-type overhang, only one side with an attached face exists. Otherwise, in the case of an H-type overhang, two or more sides with attached faces will be found. The distance measurement requires a reference plane so that the length of the overhang can be determined. This is done with a plane parallel to the Z-axis, which contains the common edge of the overhang structure with the attached face. The length is determined by the maximum-minimum distance between the points on the other side and the plane (see Fig. 6), integrating Ericson's algorithms [44].

The maximum refers to that the minimum length per point of the other side is calculated and the maximum value is taken. Our implemented distinction between H and T-type is based on the orientation of the adjacent surfaces of the beginning and end faces. The H-type overhang structure is identified by the face ends that are attached to faces, which are in this case in the opposite direction to the building direction (see Fig. 7). If only the beginning or end face has an adjacent face that points in the opposite building direction, it is a T-type. If no face points against the building direction, the object stands on the ground.

5) CURVATURE (INNER AND OUTER RADIUS)

Requirements on the inner and outer radii of structures exist in several processes, as well as in laser beam melting [14]. In general, the radii are also referred to as the curvatures of the object and they are of particular importance for the LCM method, since the method cannot print right angles. Especially curvatures below a certain radius are not possible, causing cracks at this point. In this paper, the curvature is determined by the method of Zhou *et al.* [45], which requires three points and calculates the radius of a circle using these points. Thereby, two adjacent triangles are taken and their not common vertex and a common vertex are projected on one plane (see Fig. 8). These three points are then used to calculate the curvature or radius.

6) CROSS SECTION AREA/RATIO

Maximum cross-sectional areas are to be considered in regard to the stability of the object. The area of slices is of interest as well as the ratio of the area in building direction. Even if

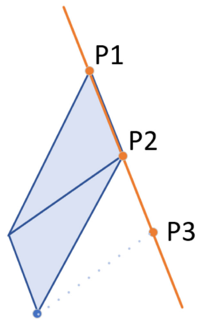


FIG. 8. **Curvature calculation with three points by Zhou *et al.* [45].**

Algorithm 1: Cross Section Area/Ratio.

Input: Triangle mesh
Output: Cross section area/ratio
1 Slice the object in building direction;
2 Calculate area of slice;
3 Calculate aspect ratio with the maximum and mininum area slice;

these guidelines are not yet considered in the LCM procedure, a solution was implemented, since these guidelines are examined in many other AM processes. The approach of this check is shown in Algorithm 1. The object is sliced into layers in the building direction. The area of each layer is calculated and the maximum and minimum area is temporarily stored for the later calculation of the ratio.

B. ONTOLOGY

An ontology is chosen for knowledge modeling because of the evolving nature of the AM technology [27]. It is still in development, having new processes introduced and new material options available. To be able to map this flood of information, it is necessary that the knowledge base can be easily updated with new information. Compared to relational databases, ontologies make it easier to add new knowledge or modify legacy data. Besides, the knowledge representation of the model in the form of ontologies enables the system to reason autonomously about the used concepts linking automatically between models, manufacturing processes and used equipment. They are also more pertinent to a tutoring system and provide richer models [8]. The knowledge base is modeled in the Web Ontology Language (OWL). The terminology and relations of the OWL language help users as well as machines to search and reuse information [26]. That makes OWL a human and machine-readable language.

The goal of this work is that the ontology represents the knowledge about the manufacturability of parts geometry. It focuses especially on representing the requirements for the manufacturability in ceramic printing with LCM [46]. The work's ontology aims also to formalize existing knowledge about the AM domain. In this context is important to consider existing work and to build on or reuse it. As mentioned earlier in the paper, the ontology by Kim *et al.* [27] includes most of the recent state of the art and offers a great starting point. However, the proposed work extends the previously mentioned one by optimizing the way the manufacturability is represented, adding the new concept of the geometry guideline. Moreover, several sources have stated that the guidelines depend on the material, machine, and machine parameter to precisely define a guideline [15], [22]. This dependence is emphasized in the ontology with new concept guidelines sets, which define the dependencies between guidelines. Both concepts geometry guideline and guidelines sets are integrated into the *Requirement* class. Besides, this work has roots in the Rosetta [47] and KnowDrift [48] ontologies, which focus on the industrial robotics domain but also modeled the concepts in a general manufacturing manner. Finally, to ensure a uniform terminology the proposed ontology uses the terminology of the ASTM 52 900 standard. The resulting ontology has the following six main classes: *Parameter, PhysicalObject, Property, Requirement, Skill,* and *Specification*. Further description of the classes follows in the next subsections. The ontology contains all necessary parts, such as materials, guidelines, machines, and processes for ceramic printing. Additionally, it includes data from the widely used Fused Deposition Modeling (FDM) process as well as guidelines from other technologies [49]. Fig. 9 presents the relevant classes and their relations for ceramic printing with the LCM process except for the guideline representation, which is explicitly described in the next section.

TABLE 1. **Resource Description Framework (RDF) Representation of a *GuidelineSet* (I.e., Each Line in the Table Represents a RDF Triple)**

Guideline set	Object property	Object
GSet_LCM	useMaterial	Alumina
GSet_LCM	useProcess	LCM
GSet_LCM	useMachine	CeraFab7500
GSet_LCM	useManufacturingPar	MP_CeraFab7500
GSet_LCM	hasGeometryGuidline	OverhangLengthTType
...	...	...
GSet_LCM	hasGeometryGuidline	AspectRatioMaxCylinder

1) REQUIREMENTS

In AM processes some *Requirements* have to be fulfilled to correctly manufacture the geometry of a part. These requirements depend on various parameters such as process, machine, manufacturing parameters and the material used. The representation of these dependencies builds the guideline set. This guideline set defines the environment and the geometry guidelines applicable to the environment. Table 1 shows an example of a guideline set for the LCM process with the material alumina, the machine Cerafab 7500, and a predefined parameter set called *MP_CeraFab7500*. *MP_CeraFab7500* is an abstraction of manufacturing parameters since more detailed information about the manufacturing parameter which influences the manufacturability is not yet available. However, the ontology already offers this possibility to include it. The set contains several geometry guidelines which are indicated

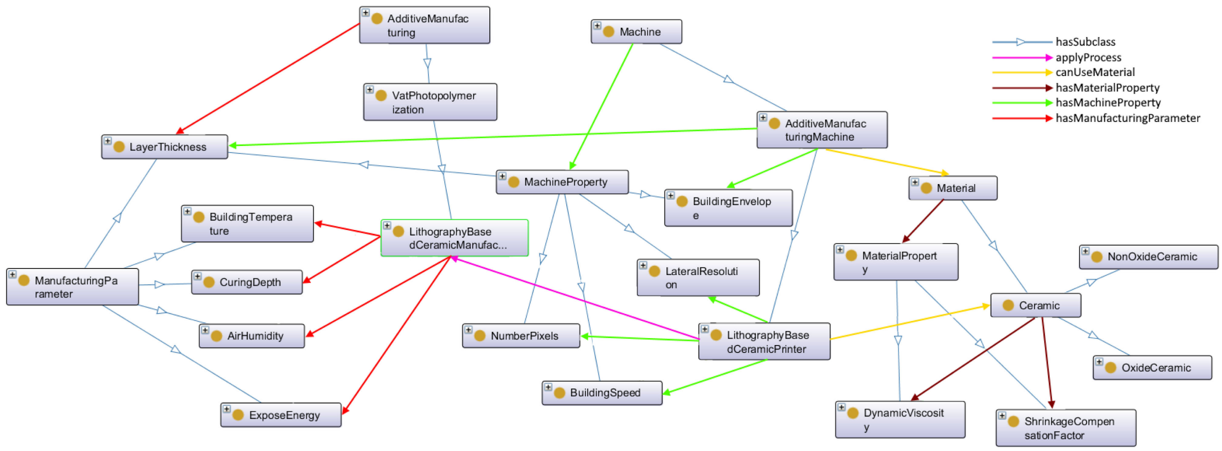


FIG. 9. Overview ontology ceramic printing with the LCM process (adopted from [9]).

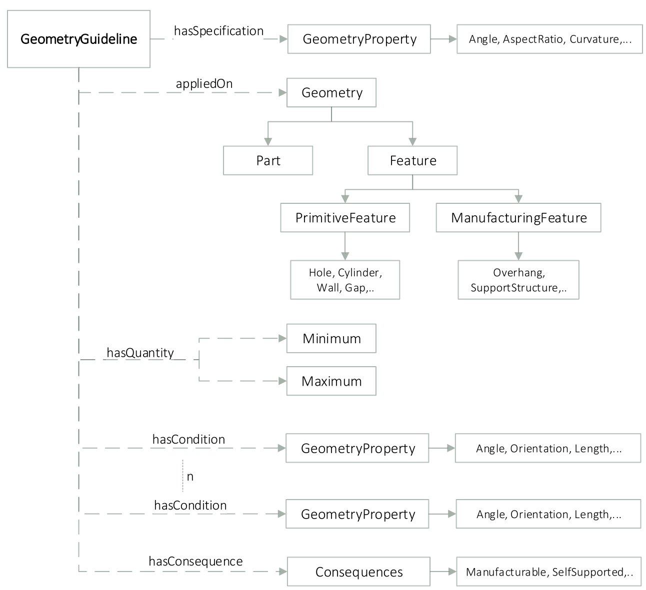


FIG. 10. Geometry guideline and its relation to other classes.

via the "..." characters. The guideline set denotation is in the table shortened to "GSet".

Guidelines are available in various forms such as, only descriptive text [16], text with illustration [13], [15], [22], or quantitative information in a table of a data sheet. This part of the ontology aims to define a uniform format with which a guideline can be defined completely uniformly. The concept and naming conventions are based on the design guideline approach by Jee *et al.* [18]. The composition of a guideline from the various components is illustrated in Fig. 10. The following example guideline illustrates this concept: A T-type (unilateral attached) overhang below the maximum length of 2.1 millimeters is self-supporting. Divided into the individual parts means that the guideline is applied to an overhang construction with the condition that is unilateral attached. The guideline should check the maximum length of the overhang.

TABLE 2. RDF Representation of a *Guideline* (I.e., Each Line in the Table Represents a RDF Triple)

Guideline	**Object property**	**Object**
OverhangLengthTType	hasSpecification	Length
OverhangLengthTType	appliedOn	Overhang
OverhangLengthTType	hasQuantity	Maximum
OverhangLengthTType	hasCondition	UnileteralAttached
OverhangLengthTType	hasConsequence	SelfSupported
Guideline	**Data property**	**Value**
OverhangLengthTType	valueDecimal	2.1

If the requirements are fulfilled, the consequences occur and the overhang would then be self-supporting.

Table 2 represents the example guideline also in RDF format. Each line in the table represents an RDF triple. Furthermore, guidelines have a consequence when they are fulfilled. First and foremost guidelines are used to check the manufacturability. However, some guidelines describe whether a feature is self-supporting. Some processes also have special post-processing steps for the guidelines to be made available to check their feasibility. In the PolyJet process, water is used to clean the part in a post-processing step [16]. So the guideline consequence is that the feature is survivable during cleaning. Fig. 9 shows the integration of the geometry guidelines in the final ontology. Especially the guidelines for the LCM process are considered, but also most other relevant guidelines are included [14], [36], [49].

2) PHYSICALOBJECT

The *PhysicalObject* class represents physical objects that are tangible in contrast to the other classes represented in the ontology. The class currently includes machines and products. The printer which executes the AM process are modeled via the *Machine* class (see Fig. 9 ($Machine \rightarrow AMMachine \rightarrow LCMPrinter$)).

3) SPECIFICATION

The *Specification* class represents some basic characteristics of things. It contains the class *Geometry* to describe the

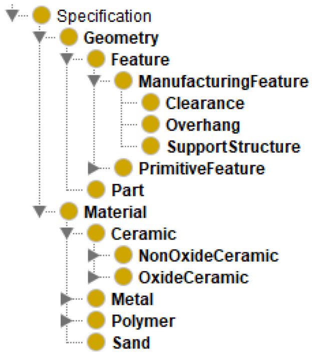


FIG. 11. **Ontology class *Specification*.**

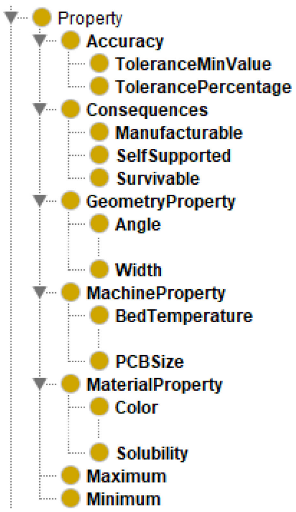


FIG. 12. **Ontology class *Property*.**

geometry of the object in form of a combination of features. It is used to describe the guideline application areas. Therefore, the geometry as a whole can also be represented as a *Part*. Moreover, the class *Material* representing the different material categories is also integrated into the concept. The focus here is on ceramics and a small overview in thermoplastic has also been integrated as illustrated in Fig. 11.

4) PROPERTY

The *Property* class represents the property of things. It is divided into *Accuracy*, *GeometryProperty*, *MachineProperty*, and *MaterialProperty* which represents the properties of their kind (see Fig. 12). Accuracy was modeled as a standalone class because it is not just machine precision; it defines the accuracy of either a machine or a process (different AM-processes have different accuracy [49]).

5) PARAMETER

The *Parameter* class represents the parameters of things. Based on some ideas from the Rosetta ontology the class is further divided into *ManufacturingParameter*, *PositionParameter*, *PostProcessingParameter*, and *WorkAreaParameter*

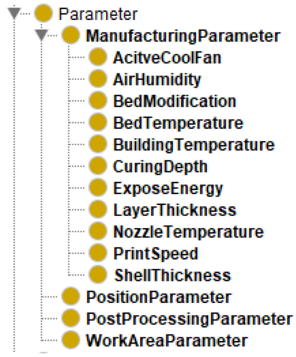


FIG. 13. **Ontology class *Parameter*.**

(see Fig. 13). The focus of the work in this class is the *ManufacturingParameter*. It contains parameters that are taken from data sheets of materials for FDM and LCM.

6) SKILL

The *Skill* class represent skills and has its roots in the Rosetta ontology. The class is divided into *AdditionalSkill*, *CompoundSkill*, and *MainSkill.AdditionalSkill* describes skills such as calibrate and parameter loading. *CompoundSkill* describes skills that are a combination of skills. The main use of a machine is represented as *MainSkill*. It contains the working process. The focus of this work lies in the *AdditiveManufacturing* class. According to the the terminology from the ASTM 52 900 standard [50] AM is divided into seven categories (see Fig. 14). Furthermore, processes belonging to the categories are also presented.

V. PROTOTYPICAL REALIZATION

This section describes the prototype of the developed framework. As shown in Section IV-A the framework consists of the Mesh Analyzer and the Ontology with a cloud platform. The cloud platform is connected with the ontology and a server on which the mesh analyzing service is allocated. It provides the user with the User Interface to the framework. The architecture is modeled in a service-oriented way to provide possibilities for flexible application scenarios: In this way, the whole system could be run locally by the engineer which designs the object and needs to check large CAD files repeatedly, but could also be deployed on the server of the 3D-printer manufacturer to conceal company sensitive algorithms or technical information about the printers. The ontology is modeled using Protégé[4] in the OWL. It is realized with a Graph Database GraphDB and accessible via a Representational State Transfer (REST) interface.

The mesh analyzer is implemented with the programming language Python in version 3.9.1.[5] The core implementation is done with the scientific computing library NumPy[6] in version

[4]Protégé available at https://protege.stanford.edu/
[5]Python available at https://www.python.org/
[6]Numpy available at https://github.com/numpy/numpy

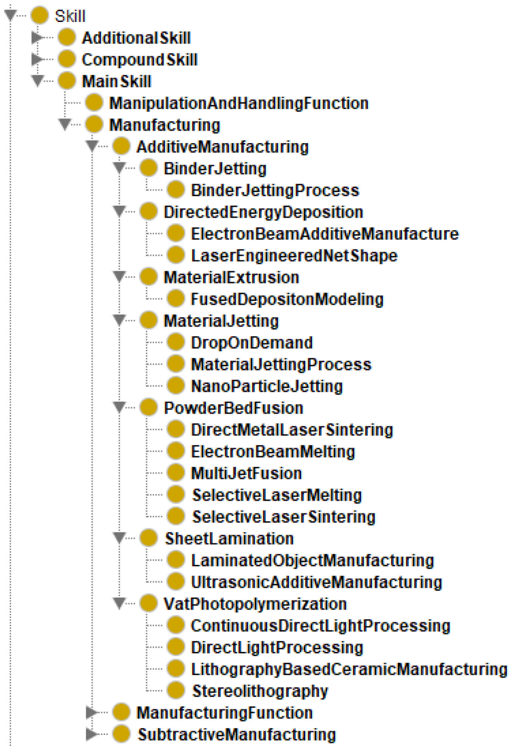


FIG. 14. Ontology class *Skill*.

1.19.5. The library PyMesh[7] in version 0.3 is used to load the STL objects as well as other geometric processes such as neighborhoods and object slicing. The mesh analyzer saves the result of the manufacturability check as a file in Polygon File Format (PLY) format. To clarify the results of the manufacturability check, visual feedback is provided. This is achieved by a visualization in the cloud platform using the mesh visualizer which is discussed below.

A. MESH VISUALIZER

The mesh visualizer prototype integrated into the cloud platform is realized with the Web Graphics Library (WebGL) framework three.js.[8] That visualizes the object with the manufacturability annotations in different color codes. Every consequence of the geometry guideline (see Section IV-B1), has its distinct color representation. This prototype represents not manufacturable faces in red and faces that need support structures in yellow. Annotations were added to highlight the exact information of the violated guideline. Ray Casting is used to select appropriate face annotations. A ray is cast through the scene at the position of the mouse that intersects the object and selects the first cut face. This face will then be colored green and an annotation box with the information of the guideline will be displayed. This box contains the names of the guideline violations and the corresponding calculated values of the mesh analyzer. Additionally, in the case of the wall thickness and gap width guideline, the opposite face is indicated and the corresponding face is colored green to make the result more comprehensible. Examples are presented below in Section VI.

[7]Pymesh available at https://github.com/PyMesh/PyMesh
[8]three.js available at https://github.com/mrdoob/three.js/

B. MANUFACTURABILITY CHECK

The procedure of the manufacturability check is illustrated in the flow diagram in Fig. 15. The cloud platform provides the user with the interaction interface to the framework and enables a user to introduce the desired model, which is usually in the STL format. In the same step, it is possible to select the appropriate AM machine available in the network based on final product specifications including also the required material and extract the guidelines required for a specific component. The ontology also offers abstract guidelines, for the case that the printer type and its manufacturing parameters are not known in advance, to be able to make a rough pre-check. The extracted guideline set is the input for the mesh analyzer. In this step, for example, the guideline with the definition of the minimum wall thickness is queried from the Ontology to be used by the wall thickness algorithm. The analyzer applies all algorithms, which are explained in Section IV-A, on the object to analyze the guidelines. The designed model can then be analyzed considering dimensions of critical areas such as thin walls, openings, small gaps, and so forth. On the other hand, the framework includes a visualizer for the designed model. The output of the analyzer is the triangular mesh with manufacturability annotations per face. This information is then visualized in the cloud platform with the mesh visualizer to give visual feedback about the manufacturability. The critical features are visualized in distinct colors, which represent different impacts and consequences.

VI. EVALUATION

This section discusses the evaluation of the manufacturability system mesh processing with the mesh analyzer. To verify its applicability, tests were carried out with different 3D-models. Besides that, also the performance in terms of speed of the algorithms is analyzed. Discovered weaknesses of the proposed approach are discussed as well. Three industrial objects are evaluated to investigate the algorithms with rather complex models. In the first place, the evaluation focuses on the recognition of the forms of the structure and not immediately on the manufacturability. Having the guideline features such as holes, pins, and overhangs must be correctly identified, the check of the manufacturability is only a comparison between the minimum feature value and the identified feature value. To illustrate different aspects of the evaluation, three different objects are presented: The first one represents a kind of a bench vice (see Fig. 16). The vice contains all relevant features for the guideline check. It contains thin walls (see Fig. 16 top left), small gaps (see Fig. 16 top right), overhangs in H-type (see Fig. 16 middle left), T-type (see Fig. 16 middle right), a little rounding (see Fig. 16 bottom left), a small channel and a large hole that need a support structure (see Fig. 16 bottom right). All features are recognized and the dimensions were determined correctly.

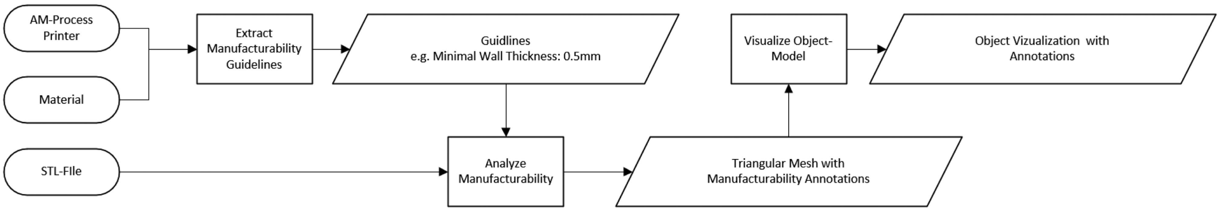


FIG. 15. Flow diagram depicting the procedure of the manufacturability check [9].

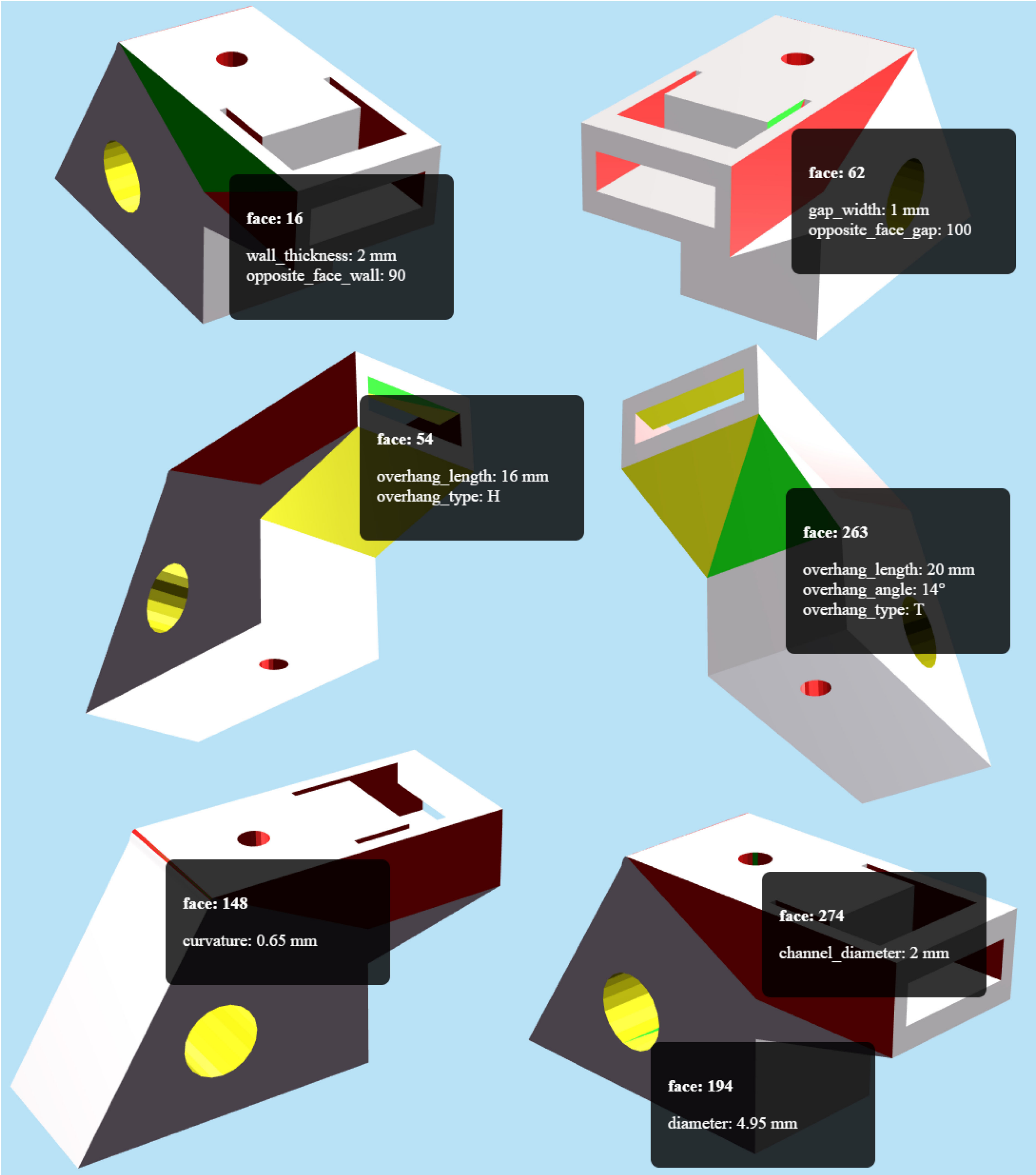


FIG. 16. Kind of bench vice including all relevant guidelines. Parts of the object which are not manufacturable are colored in red, faces that need support in yellow and faces which are selected by the user are colored in green.

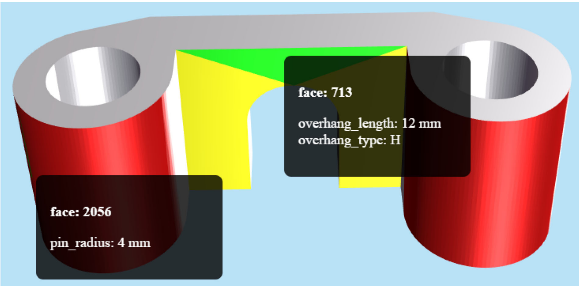


FIG. 17. Antenna mount and frame spacer for the Twin Quad Frames CHopZaw with annotations.

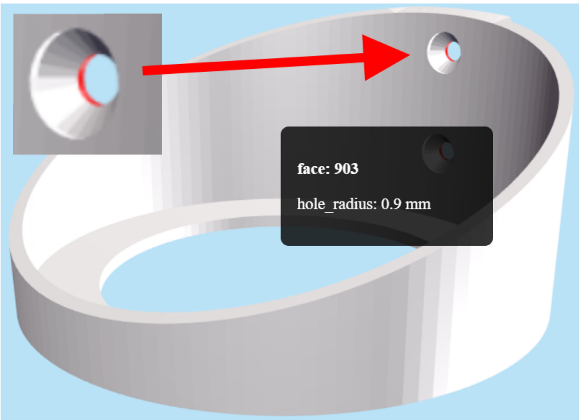


FIG. 18. Tape holder with manufacturability violation.

The second object is an antenna mount and frame spacer for the Twin Quad Frames CHopZaw (see Fig. 17). This object shows the difficulty to correctly detect features because of merging features. It contains two interlocking holes with different radii on each side which are connected by an H-type overhang. This H-type overhang contains an additional semi-circular notch. The outer sides of the holes are not completely closed cylinders which are connected by the overhang. This object illustrates the difficulty of defining a general algorithm for the different guidelines. The two internal holes are easily identifiable and calculable features. However, the surrounding cylinder is not closed but already contains the connection of the overhang (see Fig. 17). The algorithm still recognized the structure as a cylinder because it consists of two triangular planes and thus the circle can be closed. The overhang is also not easily attributable to a T- or H-type due to its shape (see Fig. 17). Also in the industrial manual check, it is not yet clearly defined whether the whole overhang is to be assigned to an H-type overhang or whether the overhang is divided into T and H-type at a certain point. The algorithm currently categorizes the overhang into an H-type and calculates the length of the complete overhang. However, the definition of the overhang and adaptations of the algorithm must be worked on.

The third object represents a mounting fixture with two holes for the attachment for tapes (see Fig. 18). It contains two small fixation holes and an uneven hole for the tape. The bottom of the bracket is also open and the rear panel has a straight edge for mounting. This object has several holes and consists of a large hollow pin and was subjected to a normal manufacturability test. It was recognized that the two inner holes have a too-small radius for manufacturing. The result is illustrated in Fig. 18 with the red hole and additionally enlarged at the upper left corner of the figure.

TABLE 3. Algorithm Runtime in Big-O Notation

Algorithm	Best Case	Average Case	Worst Case
Wall thickness Gap width naive	$O(n^2)$	$O(n^2)$	$O(n^2)$
Wall thickness Gap width k-d tree	$O(n * log(n))$	$O(n * log(n))$	$O(n^2)$
Pin/Hole radius	$O(n)$	$O(n)$	$O(n^2)$
Channel size	$O(1)$	$O(1)$	$O(1)$
Pin ratio	$O(1)$	$O(1)$	$O(1)$
Overhang	$O(n)$	$O(n)$	$O(n^2)$
Curvature	$O(n)$	$O(n)$	$O(n)$

A. ALGORITHM RUNTIME

This section analyzes the runtime of the algorithms. In computer science, the runtime is often given in the size of the instance in big-O notation. Big-O notation classifies algorithms as their runtime or space requirements grow with input size [51]. In this work, the input is a triangle mesh and the runtime is analyzed in relation to the number of triangles. Table 3 gives an overview of the runtime of the algorithms with the following parameters:

- *Wall thickness/gap width:* The naive algorithm compares each triangle with each other. This results in an algorithm runtime of $O(n^2)$ for any case. If a k-d tree is used, the comparison space is reduced. The object space is halved several times. This corresponds to a logarithmic traversing time. Besides, each triangle must be examined only with triangles in a few subspaces. In the worst case, the triangles are in all subspaces or most triangles are in one subspace that would again correspond to a runtime of $O(n^2)$.
- *Pin/Hole radius:* This algorithm searches for circular structures in the object and starts this search with the adjacent faces of each triangle. The search is aborted immediately if no circular structure can be formed. Furthermore, triangles in found circular structures are not investigated further. In the best case, there aren't any circular structures in the object at all that corresponds to a linear runtime. An object consisting only of non-closed circles would be the worst-case because it always checks the complete structure and then determines that no circle can be formed.
- *Channel size:* The channel size algorithm works with the extracted holes and checks only two faces that can be checked with constant time.
- *Pin Ratio:* The calculation of the ratio of the extracted pins is independent of the number of triangles.

- *Overhang:* This algorithm searches for overhang structures in the object and starts this search with the adjacent faces of each triangle. The runtime analysis is similar to the pin/hole radius algorithms with the same arguments related to overhanging structures.
- *Curvature:* The curvature is calculated between the adjacent faces of a triangle. A triangle has three adjacent triangles so it would correspond to an exact notation of $O(3n)$ that results in big-O notation $O(n)$.

The evaluation of the algorithms was performed on a Linux (Ubuntu) system with an AMD Ryzen 2700X 3.7 GHz computer. The runtimes of algorithms with the National Institute of Standards and Technology (NIST) test artifact which contains 7392 triangles are the following: The naive wall thickness algorithm takes 3.6 hours. The runtime of the algorithm for the wall thickness of the k-d tree needs 5 to 25 minutes depending on the dimension partitions of the k-d tree. The other algorithms need only a few seconds: Pin/Hole radius 0.42 seconds, channel size 0.6 milliseconds, pin ratio 41 milliseconds, overhang 0.1 seconds, and curvature takes 3.1 seconds.

B. DISCUSSION

The design of objects offers a high degree of freedom, which makes automatic checking using mesh processing algorithms more difficult. Previously presented approaches [7], [36], [41] mentioned that their further research would incorporate more sophisticated object designs. Rudolph and Emmelmann [36] evaluate their approach only with tested manufacturable workpieces. Shi *et al.* [41] use the NIST test artifact and little adaptions for evaluation. Tedia and Williams [7] conducted their evaluation with a bunny and a teapot. Anyhow, this work is based on the evaluation of the developed framework with rather simple objects. However, some potential improvement issues have already been identified. The shape complexity plays a very important role and can cause some difficulties to detect the necessary guideline features.

Furthermore, even after successful extraction of the features, the calculation of the desired specification such as length cannot be determined. A simple example is a screw, which has an overhang circling in height. The detection of the overhang is not a problem using the surface perpendicular, but the calculation of the overhang requires more sophisticated algorithms. In general, algorithms aimed at the recognition of feature structures are not yet applicable to all kinds of objects and must be further improved. Besides, efficient verification would be advantageous for the use of the developed framework. Due to the processing based on triangles, the algorithm runtime is strongly depending on the number of triangles. For the most time-consuming algorithm, a more efficient version has already been implemented using the k-d tree. However, the runtime is high but could be further reduced by parallel processing. Since many algorithms perform calculations with all triangles independently of each other, it is possible to execute these algorithms on a Graphics Processing Unit (GPU).

In order to meet the concrete requirements from the targeted industry and to be able to offer exact and correct results, we will further extend as well as refine and verify the ontology using formal logic to check inconsistencies and incompleteness in the developed models. Besides, studies show that most of the designers do not always consider specifics of different AM processes, treating them as a homogeneous group and guiding their decisions based on conventional processes, thereby choosing not always the best suitable but most convenient processes for producing parts [52]. Thus, the developed framework should also suggest suitable alternative AM processes or machines for printing a given artifact. Moreover, designing an efficient support structures is a significant issue in the AM processes that is based on the consideration of a range of influencing factors such as the removal process or the limits of the artifact's geometry [35]. In this context, the manufacturability framework should also incorporate a component for support generation in order to achieve better industrial acceptance and therefore usage.

VII. CONCLUSIONS AND OUTLOOK

AM makes it possible to create complex geometries in a wide variety of materials and represents competition to the conventional manufacturing processes. However, due to the novelty of the technology, all its limitations have not been fully explored. Besides, not every model that can be designed with a CAD can be manufactured using AM. To be able to print an object faultless, relevant guidelines must be considered during the design process. Guidelines depend on the used material, applied process as well as used machines and related machine parameters. Generally, guidelines should be defined and presented in a uniform way, which would simplify the application for the manufacturability check. Besides, it is of the highest importance to recognize the forms of the manufactured object, Having the correctly identified features such as holes, pins, and overhangs, the rest of the manufacturability check process is focused only on a comparison between the minimum feature value and the identified feature value.

In this work, we presented the developed knowledge-driven framework solution able to examine the geometric properties of a given design and compare it to AM guidelines. The software automatically analyzes the manufacturability of the design considering information about the used process and material. This work especially focuses on the rarely researched ceramic printing with the LCM process. In this context, an ontology is employed as a knowledge base to represent the concepts of AM and most important design guidelines for the various technologies, processes as well as material dependencies. Together with the ontology, a cloud platform, and a mesh analyzer presents the key components of the framework. The cloud platform presents the user interface of the system to the user. The mesh analyzer uses triangle-based mesh processing algorithms to recognize features and check the guidelines necessary for the LCM process. The result of the automated analysis is annotated triangle faces with the critical guidelines and their calculated values. The checked object is visually displayed with color codes for different guideline consequences, such as manufacturable or needs for a support structure. The evaluation shows that simple structures can be efficiently checked for manufacturability using predefined algorithms. However, the framework has some difficulties to recognize objects with higher complexity. Due to the unlimited possibilities of designing an object, the algorithms must be further improved to successfully extract the required structures for the guidelines checks with more complex objects.

Future work includes a more detailed definition of the guidelines for the LCM process. Especially the post-processing step plays an important role in the production of ceramic objects. This information should be mapped in the knowledge-base and checked with corresponding algorithms. Furthermore, the mesh processing algorithms with more complex geometries have to be investigated and their runtime optimized.

REFERENCES

[1] I. Gibson, D. W. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies*, vol. 17. Berlin, Germany: Springer, 2014.

[2] S. Nelaturi, W. Kim, and T. Kurtoglu, “Manufacturability feedback and model correction for additive manufacturing,” *J. Manuf. Sci. Eng.*, vol. 137, no. 2, 2015, Art. no. 0 21015.

[3] T. Lu, “Towards a fully automated 3D printability checker,” in *Proc. IEEE Int. Conf. Ind. Technol.*, 2016, pp. 922–927.

[4] P. Jaiswal and R. Rai, “A geometric reasoning approach for additive manufacturing print quality assessment and automated model correction,” *Comput.-Aided Des.*, vol. 109, pp. 1–11, 2019.

[5] W. Gao *et al.*, “The status, challenges, and future of additive manufacturing in engineering,” *Comput.-Aided Des.*, vol. 69, pp. 65–89, 2015.

[6] S. E. Ghiasian, P. Jaiswal, R. Rai, and K. Lewis, “From conventional to additive manufacturing: Determining component fabrication feasibility,” in *Volume 2A: 44th Des. Automat. Conf.*, Aug. 2018, Art. no. V02AT03A043.

[7] S. Tedia and C. B. Williams, “Manufacturability analysis tool for additive manufacturing using voxel-based geometric modeling,” in *Proc. 27th Annu. Int. Solid Freeform Fabr. Symp.*, 2016, pp. 3–22.

[8] M. Dinar and D. W. Rosen, “A design for additive manufacturing ontology,” *J. Comput. Inf. Sci. Eng.*, vol. 17, no. 2, 2017, Art. no. 0 21013.

[9] M. Mayerhofer, M. Merdan, M. Schwentenwein, and W. Lepuschitz, “Manufacturability analysis for additive manufacturing,” in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat.*, Sep. 2019, pp. 1252–1255.

[10] M. Schwentenwein, P. Schneider, and J. Homa, “Lithography-based ceramic manufacturing: A novel technique for additive manufacturing of high-performance ceramics,” in *Proc. 13th Int. Ceramics Congr. - Part B, Ser. Adv. Sci. Technol.*, vol. 88, Dec. 2014, pp. 60–64.

[11] A. Altun, T. Prochaska, T. Konegger, and M. Schwentenwein, “Dense, strong, and precise silicon nitride-based ceramic parts by lithography-based ceramic manufacturing,” *Appl. Sci.*, vol. 10, p. 996, Feb. 2020, doi: 10.3390/app10030996!.

[12] J. Wilbig, F. Borges de Oliveira, A.-F. Obaton, M. Schwentenwein, K. Rübner, and J. Günster, “Defect detection in additively manufactured lattices,” *Open Ceramics*, vol. 3, 2020, Art. no. 100020.

[13] G. A. Adam and D. Zimmer, “On design for additive manufacturing: Evaluating geometrical limitations,” *Rapid Prototyping J.*, vol. 21, no. 6, pp. 662–670, 2015.

[14] S. Lammers, J. Tominski, S. Magerkohl, T. Künneke, T. Lieneke, and D. Zimmer, “Design guidelines for a software-supported adaptation of additively manufactured components with regard to a robust production,” in *Proc. 29th Annu. Int. Solid Freeform Fabr. Symp.*, 2018, pp. 527–540.

[15] J. Kranz, D. Herzog, and C. Emmelmann, “Design guidelines for laser additive manufacturing of lightweight structures in tial6v4,” *J. Laser Appl.*, vol. 27, no. S1, 2015, Art. no. S 14001.

[16] N. Meisel and C. Williams, “An investigation of key design for additive manufacturing constraints in multimaterial three-dimensional printing,” *J. of Mech. Des. - Trans. ASME*, vol. 137, no. 11, Nov. 2015, doi: 10.1115/1.4030991!.

[17] G. A. Adam and D. Zimmer, “Design for additive manufacturing-element transitions and aggregated structures,” *CIRP J. Manuf. Sci. Technol.*, vol. 7, no. 1, pp. 20–28, 2014.

[18] H. Jee and P. Witherell, “A method for modularity in design rules for additive manufacturing,” *Rapid Prototyping J.*, vol. 23, no. 6, pp. 1107–1118, 2017.

[19] M. Mani, P. Witherell, and H. Jee, “Design rules for additive manufacturing: A categorization,” in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng.*, 2017, Art. no. V001T02A035.

[20] D. Ullman, *The Mechanical Design Process, Ser. McGraw-Hill Series in Mechanical Engineering*. New York, NY, USA: McGraw-Hill, 2003.

[21] U. Scheithauer, E. Schwarzer, T. Moritz, and A. Michaelis, “Additive manufacturing of ceramic heat exchanger: Opportunities and limits of the lithography-based ceramic manufacturing (LCM),” *J. Mater. Eng. Perform.*, vol. 27, no. 1, pp. 14–20, 2018.

[22] J. Tominski, S. Lammers, C. Wulf, and D. Zimmer, “Method for a software-based design check of additively manufactured components,” in *Proc. 29th Annu. Int. Solid Freeform Fabr. Symp.*, pp. 69–97, 2018.

[23] T. Vaneker, A. Bernard, G. Moroni, I. Gibson, and Y. Zhang, “Design for additive manufacturing: Framework and methodology,” *CIRP Ann.*, vol. 69, no. 2, pp. 578–599, 2020.

[24] S. Bechhofer *et al.*, “OWL web ontology language reference,” W3C, Feb. 2004. Online. [Available]: http://www.w3.org/TR/owl-ref/.

[25] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[26] D. Eddy, S. Krishnamurty, I. Grosse, M. Perham, J. Wileden, and F. Ameri, “Knowledge management with an intelligent tool for additive manufacturing,” in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2015, Art. no. V0 1AT02A023.

[27] S. Kim, D. W. Rosen, P. Witherell, and H. Ko, “A design for additive manufacturing ontology to support manufacturability analysis,” *J. Comput. Inf. Sci. Eng.*, vol. 19, no. 4, 2019, Art. no. 0 41014.

[28] S. Kim, D. W. Rosen, P. Witherell, and H. Ko, “Linking part design to process planning by design for additive manufacturing ontology,” in *Proc. 3rd Int. Conf. Prog. Additive Manuf.*, 2018, pp. 303–308.

[29] T. Hagedorn, S. Krishnamurty, and I. Grosse, “A knowledge-based method for innovative design for additive manufacturing supported by modular ontologies,” *J. Comput. Inf. Sci. Eng.*, vol. 18, pp. 021009–021, Mar. 2018.

[30] M. Mohd Ali, R. Rai, J. N. Otte, and B. Smith, “A product life cycle ontology for additive manufacturing,” *Comput. Ind.*, vol. 105, pp. 191–203, 2019.

[31] J. S. Liang, “An ontology-oriented knowledge methodology for process planning in additive layer manufacturing,” *Robot. Comput.-Integr. Manuf.*, vol. 53, pp. 28–44, 2018.

[32] H. Ko, P. Witherell, Y. Lu, S. Kim, and D. W. Rosen, “Machine learning and knowledge graph based design rule construction for additive manufacturing,” *Additive Manuf.*, vol. 37, 2021, Art. no. 101620.

[33] X. Liu and D. W. Rosen, “Ontology based knowledge modeling and reuse approach of supporting process planning in layer-based additive manufacturing,” in *Proc. Int. Conf. Manuf. Automat.*, 2010, pp. 261–266, doi: 10.1109/ICMA.2010.40.

[34] J. Ma, S. W. Bae, and S. Choi, “3D medial axis point approximation using nearest neighbors and the normal field,” *Vis. Comput.*, vol. 28, no. 1, pp. 7–19, 2012.

[35] M. Leary, M. Mazur, M. Watson, E. Boileau, and M. Brandt, “Voxel-based support structures for additive manufacture of topologically optimal geometries,” *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 1, pp. 1–26, Nov. 2019.

[36] J.-P. Rudolph and C. Emmelmann, “Analysis of design guidelines for automated order acceptance in additive manufacturing,” *Procedia CIRP*, vol. 60, pp. 187–192, 2017.

[37] G. Allaire, M. Bihr, and B. Bogosel, “Support optimization in additive manufacturing for geometric and thermo-mechanical constraints,” *Struct. Multidisciplinary Optim.*, vol. 61, no. 6, pp. 2377–2399, 2020.

[38] E. van de Ven, R. Maas, C. Ayas, M. Langelaar, and F. van Keulen, “Overhang control based on front propagation in 3d topology optimization for additive manufacturing,” *Comput. Methods Appl. Mechanics Eng.*, vol. 369, 2020, Art. no. 113169.

[39] W. Mycroft *et al.*, “A data-driven approach for predicting printability in metal additive manufacturing processes,” *J. Intell. Manuf.*, vol. 31, pp. 1–13, Oct. 2020.

[40] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[41] Y. Shi, Y. Zhang, S. Baek, W. D. Backer, and R. Harik, "Manufacturability analysis for additive manufacturing using a novel feature recognition technique," *Comput.-Aided Des. Appl.*, vol. 15, no. 6, pp. 941–952, 2018.

[42] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[43] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of Discrete and Computational Geometry*. Boca Raton, FL, USA: CRC Press, 2017.

[44] C. Ericson, "Chapter 5 - basic primitive tests," *Real-Time Collision Detection, Ser. Morgan Kaufmann Ser. Interactive 3D Technol.*, C. Ericson, Ed., San Francisco, CA, USA: Morgan Kaufmann, 2005, pp. 125–233.

[45] S. Zhou, J. Yin, X. Yang, and J. Wu, "Accurate curvature approximation of 3-dimension discrete points," in *Communication Systems and Information Technology* Springer, 2011, pp. 803–810.

[46] R. Felzmann *et al.*, "Lithography-based additive manufacturing of cellular ceramic structures," *Adv. Eng. Mater.*, vol. 14, no. 12, pp. 1052–1058, 2012.

[47] R. Patel, M. Hedelind, and P. Lozan-Villegas, "Enabling robots in small-part assembly lines: The "ROSETTA approach"-an industrial perspective," in *Proc. 7th German Conf. Robot.*, 2012, pp. 1–5.

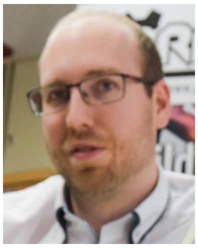
[48] M. Merdan, T. Hoebert, E. List, and W. Lepuschitz, "Knowledge-based cyber-physical systems for assembly automation," *Prod. Manuf. Res.*, vol. 7, no. 1, pp. 223–254, 2019.

[49] B. G. Ben Redwood and F. Schöffer, The 3D Printing Handbook: Technologies, design and applications, 3D Hubs, 2017, doi: 10.5555/3199991.

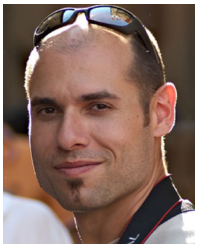
[50] "ISO/ASTM 52900:2015(en): *Additive manufacturing - general principles - terminology*," *Int. Org. Standardization, Standard*, Oct. 2015. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-astm:52900:ed-1:v1:en

[51] A. Mohr, "Quantum computing in complexity theory and theory of computation," *International workshop on quantum computation*, Illinois University at Carbondale IL, Feb. 2014.

[52] P. Pradel, Z. Zhu, R. Bibb, and J. Moultrie, "Investigation of design for additive manufacturing in professional design practice," *J. Eng. Des.*, vol. 29, no. 4-5, pp. 165–200, 2018.



MANUEL MAYERHOFER received the M.Sc. degree in visual computing from the Vienna University of Technology, Vienna, Austria, in 2020. In 2008, he joined the Practical Robotics Institute Austria, Vienna, Austria, as a Research Assistant.



WILFRIED LEPUSCHITZ (Member, IEEE) received the Ph.D. degree in electrical engineering from Automation and Control Institute, the Vienna University of Technology, Vienna, Austria, in 2018. In 2014, he joined the Practical Robotics Institute Austria,Vienna, Austria, and since 2018, he has been the Managing Director.

TIMON HOEBERT received the bachelor's degree (with Honors.) in visual computing in 2018 from the Vienna University of Technology, Vienna, Austria, where he is currently working toward the master's degree in visual computing. In 2014, he joined the Practical Robotics Institute, Vienna, Austria, as a Research Assistant.

MUNIR MERDAN received the Ph.D. degree in electrical engineering from the Automation and Control Institute (ACIN), Vienna University of Technology, Vienna, Austria, in 2009. Subsequently, he held the position of Head of Research for Cognitive Automation with ACIN. He is currently the Scientific Director of Practical Robotics Institute Austria,Vienna, Austria.

MARTIN SCHWENTENWEIN received the Ph.D. degree in chemical engineering and photopolymer chemistry from the Vienna University of Technology, Vienna, Austria, in 2012. In 2012, he joined Lithoz, Vienna, Austria, where he is currently the Head of Materials Development and R&D Coordinator.

THOMAS I. STRASSER (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from TU Wien, Vienna, Austria, in 2003. He is currently a Senior Scientist with the Center for Energy, AIT Austrian Institute of Technology, Seibersdorf, Austria. He is active as a Senior Lecturer (Privatdozent) with TU Wien. He was the recipient of the Venia Docendi (habiliation) from TU Wien in 2017. He is a Member of International IEC and IEEE standardization working groups.