FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Produktionsfähigkeitsanalyse von Additiver Fertigung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Manuel Mayerhofer, BSc.
Matrikelnummer 01328948

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Dipl.-Ing. Dr.techn. Thomas I. Strasser
Mitwirkung: Dipl.-Ing. Dr.techn. Wilfried Lepuschitz
Dipl.-Ing. Dr.techn. Munir Merdan

Wien, 1. Dezember 2019

Manuel Mayerhofer	Thomas I. Strasser

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Manufacturability Analysis for Additive Manufacturing

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Manuel Mayerhofer, BSc.
Registration Number 01328948

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Dipl.-Ing. Dr.techn. Thomas I. Strasser
Assistance: Dipl.-Ing. Dr.techn. Wilfried Lepuschitz
Dipl.-Ing. Dr.techn. Munir Merdan

Vienna, 1st December, 2019

Manuel Mayerhofer   Thomas I. Strasser

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Manuel Mayerhofer, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Dezember 2019

Manuel Mayerhofer

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung meiner Abschlussarbeit gefördert und unterstützt haben. Ich möchte mich bei meinem Betreuer Dr. Thomas Strasser für die nützlichen Kommentare, Anmerkungen und das Engagement im Rahmen des Lernprozesses dieser Masterarbeit bedanken. Mein herzlicher Dank gilt auch meinen Kollegen Dr. Wilfried Lepuschitz und Dr. Munir Merdan für ihre Unterstützung und hilfreichen Anregungen bei der Erstellung der Arbeit. Darüber hinaus bin ich Dr. Martin Schwentenwein und Dipl.-Ing. Florian Ertl von der Lithoz GmbH sehr dankbar für ihre fachliche Unterstützung zum Thema Keramikdruck. Ohne ihre Expertise wäre es nicht möglich, die Arbeit zu schreiben. Zu guter Letzt gilt mein besonderer Dank, meinen Eltern für ihre unermüdliche Unterstützung während all meiner Lebensabschnitte und vor allem für ihre Geduld und mentale Unterstützung während meines Studiums.

Acknowledgements

At this point, I would like to thank everyone who encouraged and supported me during the time of composing my thesis. I would like to express my gratitude to my supervisor Dr. Thomas Strasser for the useful comments, remarks, and engagement through the learning process of this master thesis. My sincere thanks go to my colleagues Dr. Wilfried Lepuschitz and Dr. Munir Merdan for their support and helpful suggestions in drafting the work. Moreover, I am very grateful to Dr. Martin Schwentenwein and Dipl.-Ing. Florian Ertl of the Lithoz GmbH for their knowledge support about ceramic printing. Without their expertise, it would not be possible to write the thesis. Finally, and most importantly, my warmest thanks go to my parents for their never-ending support during all my life stages and especially for their patience and mental support throughout my studies.

Kurzfassung

Die Additivfertigung hat sich in den letzten Jahren von einem Rapid Prototyping Prozess zu einem Standardprozess entwickelt. Dennoch ist es aufgrund der unterschiedlichen prozessbedingten Herausforderungen nach wie vor kein weit verbreitetes Herstellungsverfahren. In diesem Zusammenhang ist es von entscheidender Bedeutung, die Herstellbarkeit der konstruierten Teile zu überprüfen. Es gibt immer mehr druckbare Materialien. Die Entwicklung von neuen Verfahren ist ständig in Bewegung. Diese Arbeit konzentriert sich auf den wenig erforschten Keramikdruck mit dem Lithographie basierenden Keramikherstellungsverfahren (LCM). Es wird untersucht, wie Designempfehlungen definiert werden. Darüber hinaus wird erläutert, wie allgemeine Konstruktionsrichtlinien in maschinenlesbares Format gebracht werden können, um eine einfache Abfrage der erforderlichen Informationen für einen automatischen Produktionsfähigkeitsüberprüfer zu gewährleisten. Diese Arbeit stellt ein wissensbasiertes Framework vor, das in der Lage ist, geometrische Eigenschaften eines Teils automatisch zu untersuchen und mit den Richtlinien für die additive Fertigung zu vergleichen. Als Wissensbasis wird eine Ontologie verwendet, die Informationen über die Produktionsfähigkeiten von additiven Herstellungsverfahren, Druckern und Materialien enthält. Das System für die additive Fertigung verwendet dreiecksbasierte Polygonnetz-Verarbeitungsalgorithmen, um Merkmale zu erkennen und die für den LCM-Prozess notwendigen Richtlinien zu überprüfen. Die Funktionalität sowie die Performance in Form der Laufzeit der Algorithmen werden mit Testwerkstücken, dem National Institute of Standards and Technology(NIST) Testartefakt und einigen industriellen Objekten evaluiert. Die Evaluierung zeigt die Machbarkeit und ihre Grenzen der Produktionsfähigkeitsanalyse.

Abstract

Additive Manufacturing (AM) evolved in the last years from being a rapid prototyping process to a standard manufacturing process. Nevertheless, it is still not a widely used manufacturing method due to different process-related challenges. In this context, it is of vital importance to inspect the manufacturability of the designed parts. The possibilities of printable materials are becoming more and more. New processes are constantly being developed and this thesis focuses on the not yet widely researched ceramic printing with the Lithography-based Ceramic Manufacturing (LCM) processes. It is examined how design recommendations are defined. Furthermore, how to map general design guidelines into a machine-readable format to ensure an easy way to query the required information for an automatic manufacturability checker. This thesis presents a knowledge-driven framework able to automatically examine geometric properties of a part and compare it to additive manufacturing guidelines. As a knowledge base, an ontology is used which contains information about the capabilities of additive manufacturing processes, printers, and materials. The additive manufacturing manufacturability system uses triangle-based mesh processing algorithms to recognize features and check the guidelines necessary for the LCM process. The functionality, as well as the performance in the form of the runtime of the algorithms, are evaluated with manufacturable workpieces, the National Institute of Standards and Technology (NIST) test artifact and some industrial objects. The evaluation shows the feasibility of manufacturability analysis and its limitations.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Problem Description

AM, also referred to as 3D printing offers more and more potential in comparison to conventional manufacturing methods like milling, casting or turning, due to the fast development of printer technologies and printable materials [GRS+14]. The advantages of AM technologies are the faster turnaround time and the feasibility of producing complex geometries with different materials [NKK15]. The costs of 3D printers have fallen dramatically. This leads to the fact that an increasing number of people design individual object models as a hobby and make them available on platforms like thingiverse [thi, Lu16]. However, the wider usage of additive manufacturing for end-use part production is still limited, since different process-specific challenges such as rough surfaces or the stair-stepping effect caused by the layer-by-layer manufacturing harm the industrial application [LDAZ16]. Besides, often the limitations arise in the form of minimum feature size producible by the process [JR19]. Moreover, not every model that can be designed with a Computer-Aided Design (CAD) software can be manufactured using 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 like AM process, printer resolution, layer thickness, the material used and so on [GZR+15]. Nevertheless, engineers and users of additive technologies often lack the awareness of manufacturing considerations leading to lower quality parts and print failures. In this context, extensive knowledge and understanding of additive manufacturing constraints and restrictions are needed for industrial adoption of the technology for end-use part production [GJRL18]. At present, CAD software does not support assistant designing for AM. Users have to look for design guidelines, which provide information on how to design a specific geometry so that it is printable, in the literature or 3D printing service platforms like 3D Hub [3dh]. This process is time-consuming and also difficult to get an overview of the diverse guidelines. Therefore, AM service providers get a huge number of unprintable object models. These object models have to be checked manually by experts for their manufacturability [TW16]. Besides, in use are a variety of AM technologies with different AM processes, materials and printer types with diverse capabilities and constraints (for example surface finish and accuracy). Generally, a system is required that can support engineers through the design process of a specific model and configuring process parameters [DR17]. These requirements imply the 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.

1.2 Research Question and Objectives

This work aims to design a prototype overall system for manufacturability analysis in AM with the focus on the LCM process. The work should answer the questions: How should the knowledge base be mapped in the domain AM to make it expandable for different processes and to ensure a simple query of the required information for an automatic manufacturability check? Which guidelines have to be checked at the LCM process and which possibilities are there to check them?
This thesis improves the automation of analyzing manufacturability by two parts. The first part, which leads to an improvement of the automation process, is the development of an ontology about the AM technologies and, dependent thereon, manufacturability requirements. This is intended to represent the most important design guidelines for the various technologies as well as material dependencies. The second part consists of a software system that checks the manufacturability of the object models with the information of the ontology as input. The system should help AM service providers to reduce the number of non-printable object models by performing a pre-process manufacturability check. There the designers can evaluate their object models and get visual and textual feedback about the manufacturability with the specific AM process and material. This work presents a knowledge-based framework that considers specific restrictions of the AM process and automatically analyzes the manufacturability of a particular object model.

1.3 Structure of the thesis

This thesis is structured in the following chapters:

- **Chapter 2 Background and Related Literature:** In this chapter the fundamentals are presented which are necessary for an Additive Manufacturing Manufacturability Analysis System (AMMAS). The state of the art of design guidelines, ontologies and mesh processing concerning AM will be explored. In particular, an overview of the manufacturability mesh processing approaches is provided. The unsolved problems of the state of the art are listed and solutions are presented.

- **Chapter 3 Additive Manufacturing Manufacturability Analysis System:** Chapter 3 covers the core part of the work. The AMMAS and the functionality, as well as the concrete prototypical implementation, are presented here. First, an overview of the system is provided with the sequence of the system processes. Second, the mesh analyzer is presented and its algorithms described and illustrated. The third part covers the knowledge base in the form of an ontology and describes the concepts and structure of it in detail. The last section presents the prototypical implementation, especially the mesh visualizer and the data exchange format Polygon File Format (PLY).

- **Chapter 4 Evaluation:** The evaluation chapter shows the evaluation of the mesh processing algorithms in different aspects. The first aspect is the variety of objects. The evaluation is carried out with test manufacturable workpieces, as well as the NIST test artifact and industrial objects. Furthermore, the curvature depending on the triangle resolution and the algorithm runtime depending on the number of triangles is investigated. Shortcomings of the approach conclude the chapter.

- **Chapter 5 Conclusion and Future Work:** This chapter give an overview of the work and its contributions, as well as future work.

CHAPTER 2

Background and Related Literature

Significant efforts have been made to verify the manufacturability of a given model [GZR+15]. Manufacturability analysis has been researched through various aspects. In recent years more and more printer technologies and possible printable materials emerged. As a result, there are constantly changing demands on the print process. A large number of design recommendations for various printing technologies are created. Generally, valid requirements are defined as guidelines. Some recommendations and general guidelines will be published. Section 2.1 provides an overview of various design guideline publications and approaches to generalize them. The definition of guidelines is the first step that is necessary to determine the manufacturability. As a second step, this information must also be encoded in a machine-readable manner. Ontologies allow representing existing knowledge systematically in a human and machine-readable format [BvHH+04]. There are already some approaches that use ontologies in the area of AM that are described in Section 2.2. For an automatic manufacturability analysis, it is not only necessary to represent the information but also to be able to check them on a part's geometry. Research has also been done in this area to develop methods that extract and analyze features for manufacturability from the part's geometry. These approaches are presented in Section 2.3. With the existing state of the art, it is not possible to solve all current problems. The unsolved problems are highlighted in Section 2.4 and the proposed solution of the problems is discussed.

2.1 Design Guidelines in Additive Manufacturing

AM offers high design freedom. This makes it possible to produce complex structures, such as lattice structures or bionic and topologically optimized structures [LTM+18]. However, AM can not print every model. It has to deal with process, material and machine parameter specific restrictions. The geometry requirements are different from those of conventional manufacturing processes and the necessary information is not yet available in an applicable context [AZ15]. As AM is a new technology and new AM processes are constantly being developed, not all constraints and dependent variables have been investigated yet. Design recommendations are created to make efficient use of AM. These design recommendations support the design for production to enable robust manufacturing. The restrictions are determined by experiments and documented in various forms. In recent years this has resulted in a large number of process-specific recommendations with different qualitative and quantitative information. That makes it time-consuming and difficult to get an overview of the recommendation which is needed for their specific purpose.

Lammer et al. [LTM+18] prioritize the huge number of design recommendations into four levels:

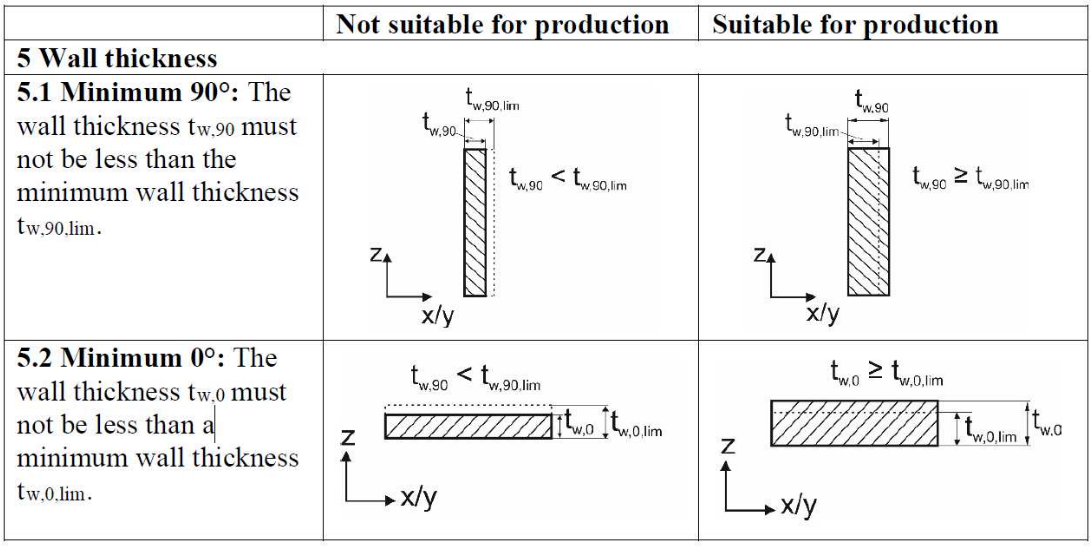
1. Endangers the manufacturability or functionality of the component
2. Loss in component quality
3. Determination by the user without influence on manufacturability or component quality
4. Already taken into account in other guidelines

The first priority level contains guidelines in which violation of the guidelines endangers the manufacturability or functionality of the component. Typical guidelines are minimum wall thickness, hole radius, or maximum overhang length.
The second priority level contains guidelines with an influence on the manufactured component quality. If these guidelines are violated, the component can be manufactured and maybe fulfill its intended task but will show deteriorations in quality. One example is the outer radius which leads to the staircase effect in different AM processes.
The third priority level contains guidelines that give hints on specific design features without quantifiable indications. Examples are the choice of the positioning of the part in the building chamber or providing openings to remove powder in cavities.
The fourth priority level contains guidelines that are already considered in priority levels one to three but with another name. For example minimal gap width and minimal space between features.
For the implementation of an automated manufacturability checker, the priorities level one and two are initially of interest. To give a brief overview of which design guidelines fall into this levels, Lammers et al. [LTM+18] has summarized the guidelines. They are focusing on guidelines for the laser beam melting AM process, but the guidelines for the process contain concepts that also apply to other AM processes.

The guidelines are categorized into eight categories:

- Surface orientation
- Outer radius
- Inner radius
- Cross-sectional area change in building direction
- Wall thickness
- Gap width
- Overhang
- Cross section area

The category wall thickness is exemplary shown in Figure 2.1.



	Not suitable for production	Suitable for production
5 Wall thickness		
5.1 Minimum 90°: The wall thickness $t_{w,90}$ must not be less than the minimum wall thickness $t_{w,90,lim}$.	$t_{w,90} < t_{w,90,lim}$	$t_{w,90} \geq t_{w,90,lim}$
5.2 Minimum 0°: The wall thickness $t_{w,0}$ must not be less than a minimum wall thickness $t_{w,0,lim}$.	$t_{w,90} < t_{w,90,lim}$	$t_{w,0} \geq t_{w,0,lim}$

Figure 2.1: Wall thickness guideline [LTM+18]

The AM manufacturability recommendations also depend on the material and intended use. Each material has different properties and therefore has different printing possibilities. Kranz et al. [KHE15] investigated in a study about design guidelines for laser AM of lightweight structures with the material Ti6AI4V. It is a high-strength titanium alloy consisting of titanium, six mass percent aluminum, and four mass percent vanadium. The alloy is by far the most commonly used titanium alloy and is characterized by high strength and good resistance to corrosion. Experiments were used to define guidelines, which are made available in quantitative and qualitative data in tabular form. Wall thickness, gap width, holes with different radii are investigated in different part positions and orientations to determine the process limits. One example of their investigation is shown in Figure 2.2, which shows the qualitative explanations and restrictions in a quantitative way. The example shows recommendations for walls, cavities, and material distribution.

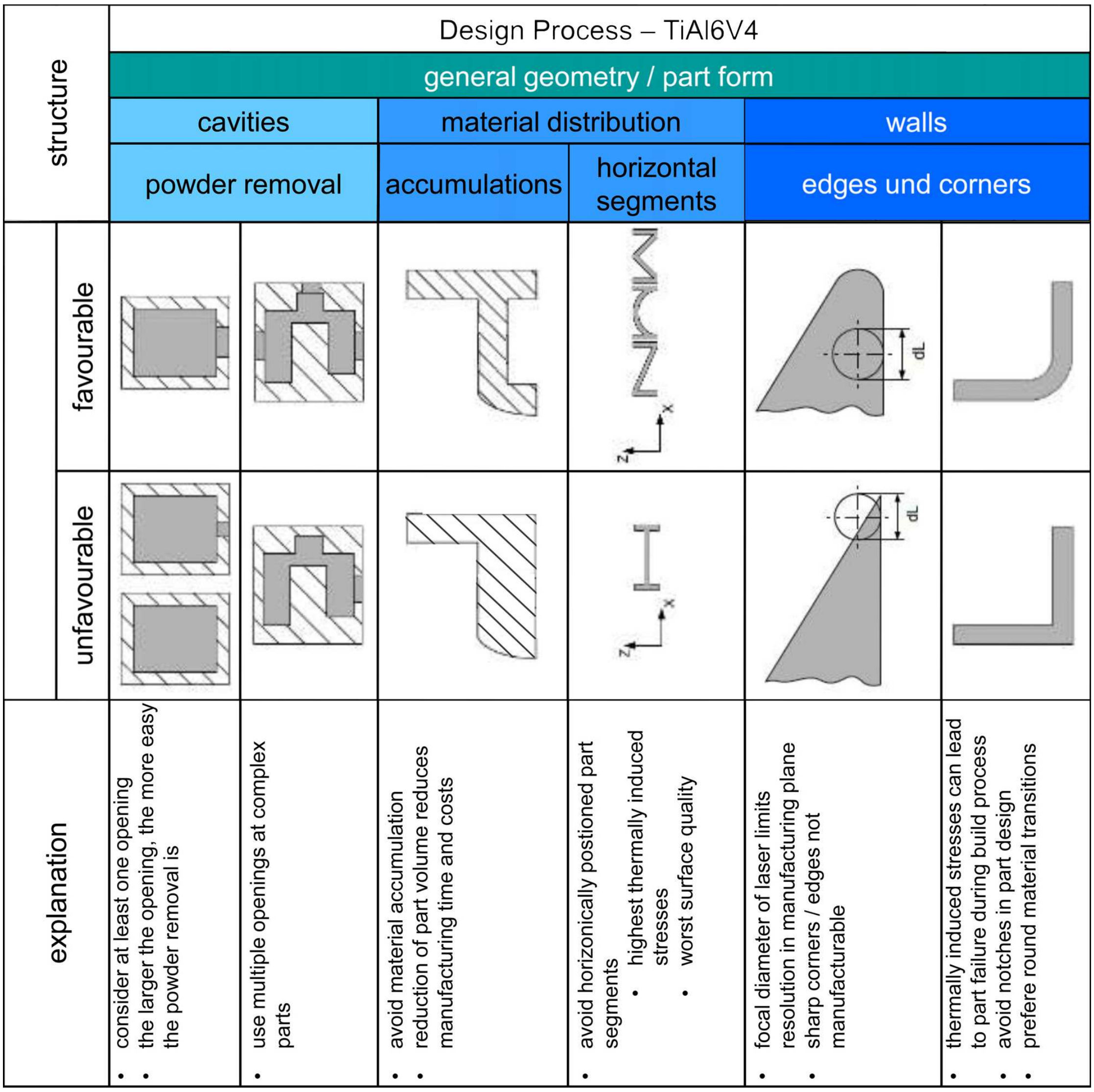


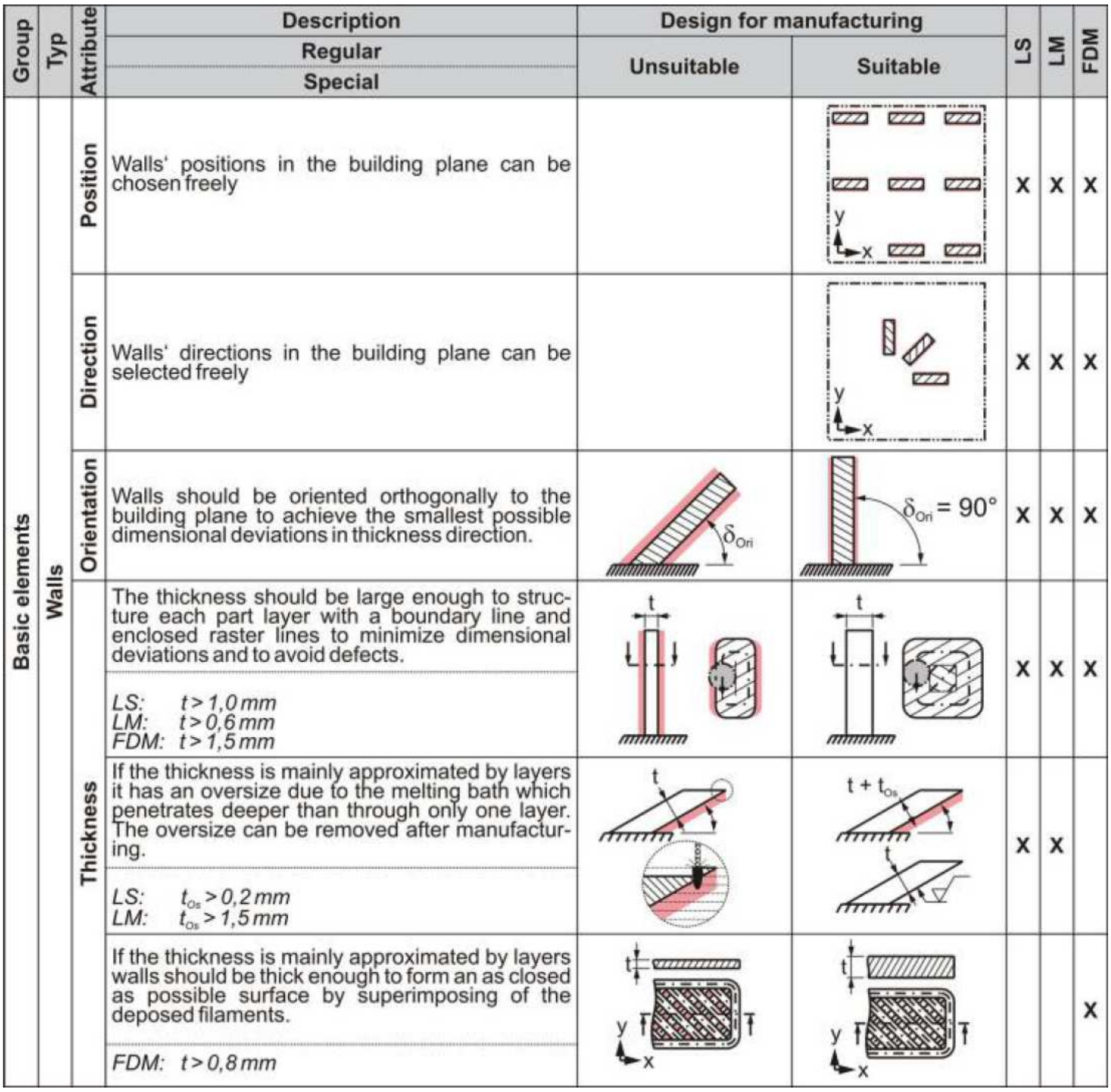
Figure 2.2: Design guideline laser AM TiAI6VA example [KHE15]

Meisel et al. [MW15] investigates in quantitative design thresholds for PolyJet material jetting. PolyJet offers the possibility to print the part in several materials at the same time. In addition, the materials can also be mixed during printing, which leads to

completely new material properties. The low layer thickness of only 16 - 32 μ m leads to a high level of detail and enables the design of complex structures. However, possible complexity also entails some limitations. The investigation of the restrictions by Meisel et al. [MW15] identifies four key manufacturing constraints. The minimum feature size, which is a process constraint used in every AM technology. The minimum self-supporting angle and support material removal, which are important by processes that use support material. For example, binder jetting and selective laser sintering do not need support structures since the powder acts as a support when the object is built up. Last, a special material jetting constraint is feature survivability during cleaning. The PolyJet uses water to clean the part in a post-process step. The minimum survivable feature can be larger than the minimum printed feature.
Usually, guidelines are examined depending on the process, as shown above for Laser AM [KHE15] or Material Beam Method [MW15]. However, there are also approaches to consider design guidelines more globally and to investigate commonalities. The research project Direct Manufacturing Design Rules (DMDR) [DMR19] has developed guidelines for AM for users in science, industry, and education. They developed a process independent method for the development of design rules. Design guidelines were not examined for parts but for standard elements, which are subdivided into three categories [AZ15]:

- Basic elements: Elementary geometrical shapes (e.g. walls and cylinders).
- Element transitions: Areas in which basic elements are combined with each other (e.g. joints).
- Aggregated structures: Arrangements of two or more basic elements and its element transitions (e.g. overhangs).

Based on this method, design rules for the additive manufacturing processes laser sintering, laser melting, and fused deposition modeling were developed. 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 rule catalogue [AZ14, AZ15]. An excerpt from the catalog is shown in Figure 2.3. 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) [DMR19] was started to extend the scope of the previously developed guidelines. Information about the project has not yet been published.



Group	Typ	Attribute	Description Regular Special	Design for manufacturing: Unsuitable	Design for manufacturing: Suitable	LS	LM	FDM
Basic elements	Walls	Position	Walls' positions in the building plane can be chosen freely			X	X	X
		Direction	Walls' directions in the building plane can be selected freely			X	X	X
		Orientation	Walls should be oriented orthogonally to the building plane to achieve the smallest possible dimensional deviations in thickness direction.	δ_{Ori}	$\delta_{Ori} = 90°$	X	X	X
		Thickness	The thickness should be large enough to structure each part layer with a boundary line and enclosed raster lines to minimize dimensional deviations and to avoid defects. *LS: t > 1,0 mm* *LM: t > 0,6 mm* *FDM: t > 1,5 mm*	t	t	X	X	X
			If the thickness is mainly approximated by layers it has an oversize due to the melting bath which penetrates deeper than through only one layer. The oversize can be removed after manufacturing. *LS: t_{Os} > 0,2 mm* *LM: t_{Os} > 1,5 mm*	t	t + t_{Os}	X	X	
			If the thickness is mainly approximated by layers walls should be thick enough to form an as closed as possible surface by superimposing of the deposed filaments. *FDM: t > 0,8 mm*	t	t			X

Figure 2.3: Design catalogue [AZ15]

Design guidelines are carried out by experimenting with specific machines and corresponding parameter settings with different materials. Usually the guideline description is only textual (compare [MW15], [KHE15]) and is slightly different from different sources. This leads to ambiguities and makes it difficult to have a consistent vocabulary for AM design solutions. Jee et al. [JW17] propose a formal design rule representation to standardize guidelines modularly. The representation looks like the following structure:

Category (type), if {Conditions} then {Consequences};

If this is about this category (feature with special properties) and these conditions are present, these consequences occur. One example rule is defined by Jee et al. [JW17] 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." The category is overhang with the specific type angular. The rule comprises of three conditions. First, the angle is greater than 45 degrees. Second, it is an undercut angle and third, it is built by a metal-based powder bed fusion process. The consequence is that there is no need for a support structure. It is self-supporting.
In addition, they introduce the concepts of primitives and modules for their representation. A primitive represent a physical feature parameter or variable like *angle*, *radius*, or *raw material type*. Furthermore, it can be divided into three categories geometry, processes, and materials [Ull10] (see Table 2.1). A module represents an implicit design feature like *support structure* or *overhang* and consists of a set of primitives, modules or a combination of both. Primitives and modules are process independent defined. To show the applicability, Jee et al. [JW17] did two case studies in which they redefined preliminary guidelines with constraints and associated parameters into their design rule representation. They conclude that the use of modular components helps to easier interpret and implement guidelines.

Geometry (G)	**Process/Machine (P)**	**Material (M)**
Part dimension	Platform orientation	Type
Part location	Platform dimension	Thermal property
Part orientation	Platform location	Physical property
Part tesselation tolerance	Build power property	
Feature dimension	Build power type	
Feature location	Build tool scale	
Feature orientation	Build tool offset	
Feature shape	Build tool location	
Feature topolgoy	Build tool speed	
Feature property	Build area	
Feature undercut angle	Build layer thickness	

Table 2.1: Design fundamentals [MWJ17], primitives [JW17] common to different AM processes

Another approach to efficiently represent design guideline is proposed by Mani et al. [MWJ17]. 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. Figure 2.4 illustrates the methodology. DF can be categorized into geometry related parameters, material parameters, and machine parameters related to Ullman et al. [Ull10]. Examples of DF are presented in Table 2.1. By definition of the authors Mani et al. [MWJ17], DPs are logical correlations capturing process parameter and control parameters and DR's are specific correlations that provide needed insight into manufacturability. The concept makes it possible to encode new knowledge formally, consistently and expandable.

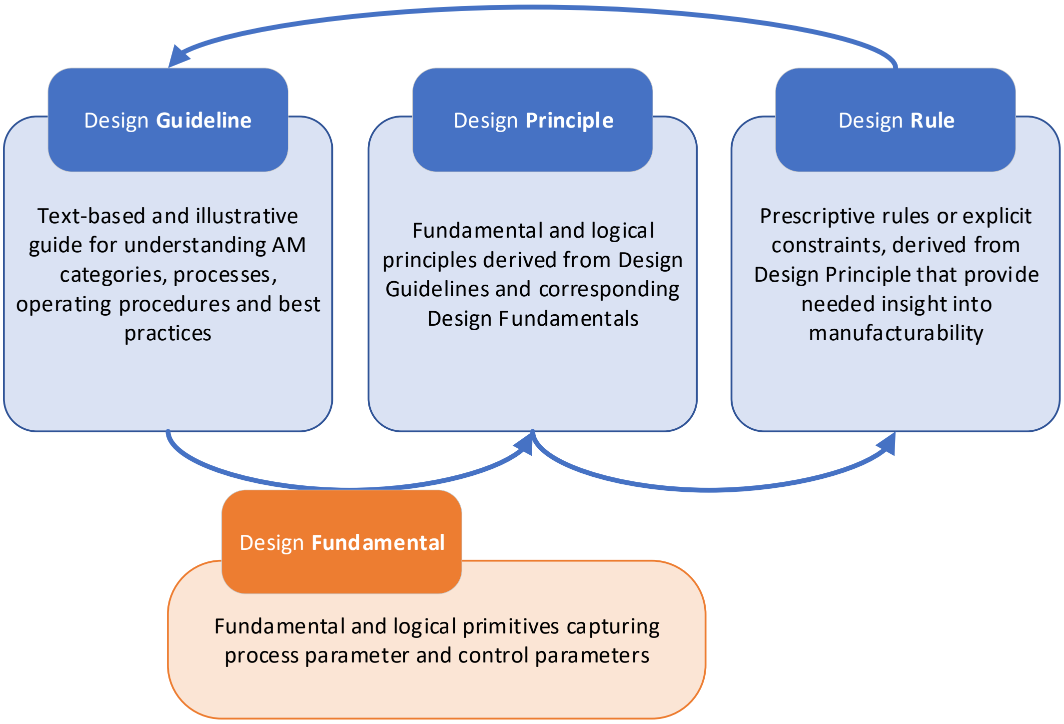


Figure 2.4: Guide-to-Principle-to-Rule methodology [MWJ17]

The state of the art analysis showed that there is a lot of research in the area of design guidelines, recommendations and rule sets. Most approaches specify special AM technologies and their constraints, but there are also similarities as shown by Adam et al. [AZ15]. Besides, there are several points of view on the manufacturability checks that are ultimately the same (see priority level 4 [TLWZ18]). General feature constraints like minimum feature size and constraints regarding feature categories like walls, gaps, holes, and pins are to be considered in every technology. However, there are also specific geometry requirements for each process (see material jetting [MW15]). As well as requirements that are not necessary with certain technologies such as overhangs with binder jetting and selective laser sintering. About the LCM procedure there exists own work that deals with explicit guidelines. However, Scheithauer et al. [SSMM18] have described the opportunities and limitations in their work. Furthermore, two methods ([JW17, MWJ17]) were presented to encode the knowledge formally, consistent and AM process independent and dependent. These methods are based on a rule set with mathematical functions. Another way to formally define knowledge is to define ontologies, which will be discussed in more detail in the next section.

2.2 Additive Manufacturing Ontology

With the development of the semantic web in recent years, a possibility emerged that allows representing the existing knowledge in a systematic way for making information context searchable and functional for its intended use [BvHH+04]. 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 [Gru93, EKG+15]. This type of information processing has already been investigated by various sources under the term Design for Additive Manufacturing (DFAM) [DR17, HKG18, KRWK18, KRWK19]. DFAM is a general design method whereby functional performance and/or other key product life-cycle considerations such as manufacturability, reliability, and cost can be optimized subject to the capabilities of additive manufacturing technologies [TZ16]. Due to the reusability of ontologies, it is standard to fall back on previous models for new developments [EKG+15]. Therefore, the current state of the art can be described very well with Kim et al. latest work [KRWK19]. The ontology provides an updated DFAM knowledge based on Kim et al. previous work [KRWK18] and also includes the research from Dinar et al. [DR17]. Their work incorporates two concepts that are already described in Section 2.1. The GPR methodology [MWJ17] which is used for principles of design rules in AM (American Society for Testing Materials (ASTM) WK54586) and the module concept by Jee et al. [JW17]. They try to formulate the ontology as general as possible in the field of AM. However, they focused first on the powder bed fusion process, because the design guidelines in this field are well researched and have high complexity in comparison to other AM processes [KHE15]. Kim et al. [KRWK19] divided the knowledge into two parts: Part design and process planning (see Figure 2.5). The part design consists of design features and geometry parameters. Process planning contains information about the material and process parameters. The work defines a concept which links part design with process planning. Between these parts, they have inserted a concept of manufacturing features that combines the design perspective with the production with a specific process and AM capabilities. In other words, the designer should consider during product design the manufacturability of the design features for AM. This means, for example, that an inclined wall forms an overhang and must be applied to special guidelines. Their ontology development ultimately resulted in a structure which has five top-level classes [KRWK19]:

- Feature
- Parameter
- AM_Capability
- AM_Process
- Machine

The class *Feature* represents design and manufacturing features and provides geometric entities necessary for manufacturability analysis. The *Parameter* class includes parameters related to features, materials, and processes. *AM_Capability* is used to represent the capabilities of AM machines and processes to form design guidelines. The classes *AM_Process* and *Machine* which contains information categorization to AM processes and machines.
The manufacturability analysis in the ontology is performed by reasoning with the Semantic Query-enhanced Web Rule Language (SQWRL). To demonstrate their effectiveness they performed two case studies. Kim et al.[KRWK19] conclude that it is possible to represent and query guidelines with their system but the information structure is complex and the construction of the SQWRL rules is complicated.

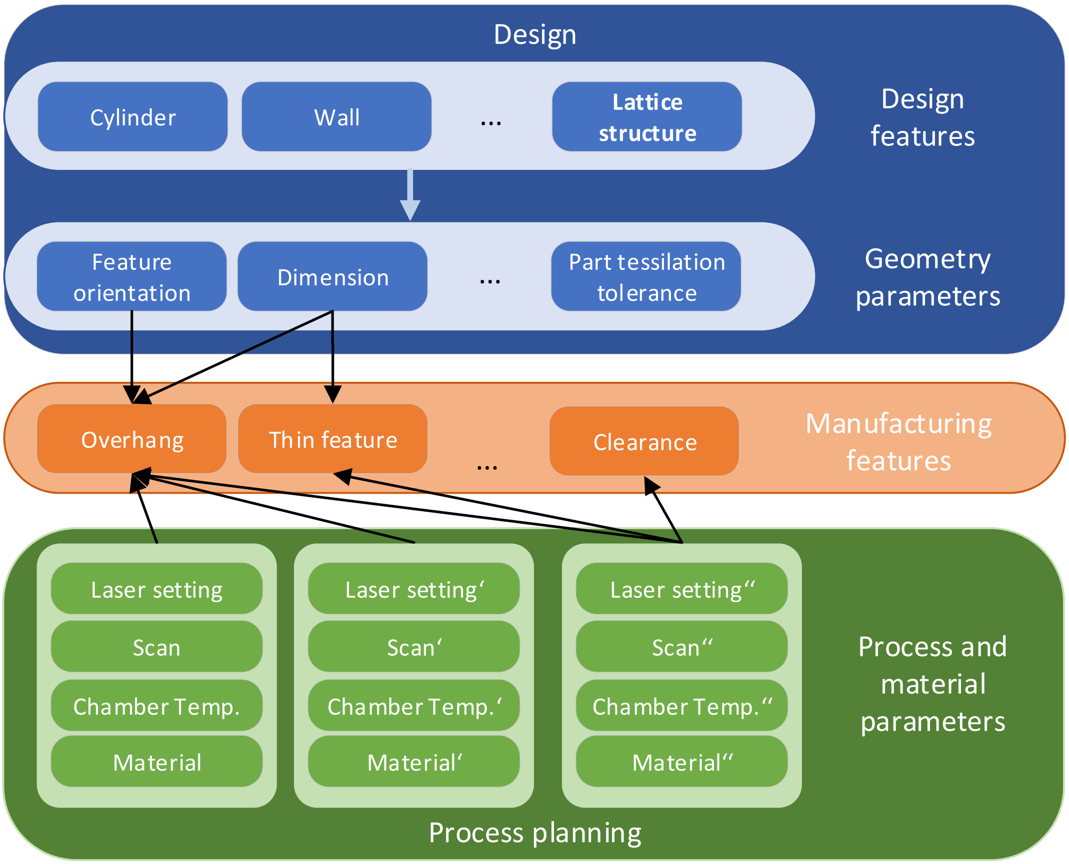


Figure 2.5: Overview DFAM ontology [KRWK19]

Further, Hagedorn et al. [HKG18] proposed a a ontology for Innovative Capabilities of Additive Manufacturing (ICAM). The work has the aim to facilitate innovative use of AM. They include past knowledge from the manufacturing domain and enterprise domain in relation to past AM products. This gives the opportunity to query for designed products based on past successes. It provides a formal description by linking knowledge from business with knowledge bases about the capabilities of AM processes and machines. The ontology provide knowledge for the designer's specific product needs.

2.3 Manufacturability Mesh Processing

To determine the manufacturability, the geometry of the object to be printed must be examined. For this feature recognition methods are used. Feature recognition is a sub-discipline of solid modeling that focuses on the design and implementation of algorithms for detecting manufacturing information from three-dimensional solid models produced by CAD systems [HPR00]. Automatic feature recognition has been active research where many different techniques are proposed. There is a summary of methods taken from [SZB+18] of feature recognition methods which state of the art automated manufacturability analysis approaches build on:

- Graph matching
- Volume decomposition
- Rule-based
- Neural network based
- Hybrid approaches

Feature recognition is not sufficient at all for the manufacturability analysis. Many possible interpretations cause algorithm complexity and lead to expensive computations. Furthermore, some methods have a loss of topological and geometric information which are necessary for the manufacturability analysis. For this reason, not every recognition method can be used. In the area of automated manufacturability analysis, some approaches have been presented in the last years. The methods interpret the solid model in terms of predefined features like 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.

2.3.1 Two Dimensional Object Model Analysis

AM manufactures the solid model in a layer-wise process. Each layer should be printed without errors. Therefore, one possibility is to analyse the manufacturability in two dimensions at each layer. In AM a layer is called slice. A slice is obtained by intersecting the three-dimensional model by a two-dimensional plane in pre-defined height depending on the printer resolution, normal to the build direction. There are different ways to analyze certain guidelines.

Medial Axis

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 [MBC12]. There is also the possibility to use balls in three-dimensional space. In two-dimensional space, several circles are layed into the component so that they touch the walls at least at two points without intersecting the walls or touching each other (see Figure 2.6). The medial axis results by linking the centers of the circles to form a line. Wall thickness and curvature radii are determined by the radii of the maximum circles. This works well for structural part members ('inside') as well as for cavities, e.g. gap widths ('outside'). Under consideration of the angles between the normal vectors at the boundary points of the maximum circles, it can distinguish between radii and thickness measures.

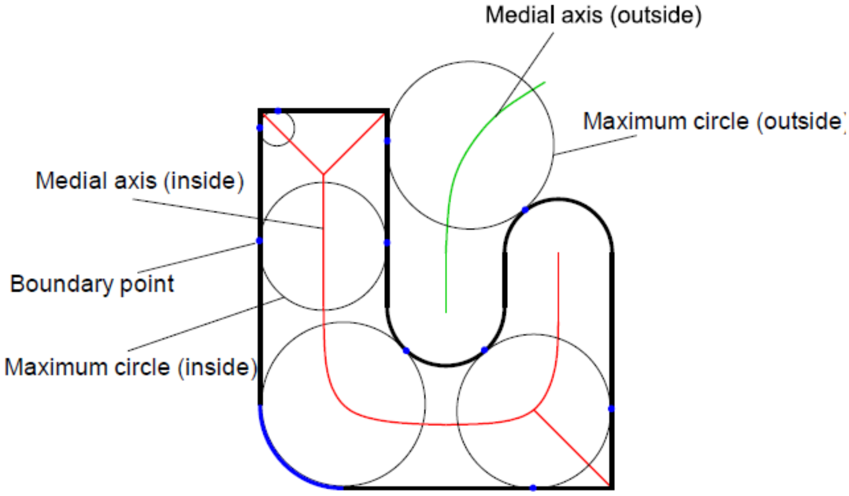


Figure 2.6: Medial axis of a component inside and outside. The maximum circles can be used to determine radii and distances. Red line medial axis inside the component. Green line outside the component [TLWZ18].

Shape Diameter Functions

Shape Diameter Functions (SDF) are widely used for the estimates of local thickness of shapes. Jaiswal and Rai [JR19] uses a variation of SDF for segmenting thin regions in slices. The SDF function maps to each point on the boundary of the sliced object the local diameter. The slice consists of multiple polygons that represent the shape of the object. Each polygon is defined by an ordered list of boundary vertices and edges that connects them. To get a finer resolution of the SDF map the polygon edges are further resampled to create new vertices between two existing ones. The computation of the SDF works as follows. Twelve rays are sent inwards at an angle of [-15,+15 degree] around the inward normal. The distance at the intersection with the boundary is recorded and then the standard deviation and median are calculated. The SDF value of the vertex is then calculated from the mean value of the distance by a maximum of one standard deviation from the median. The result of the SDF is shown on a slice of a cactus in Figure 2.7.

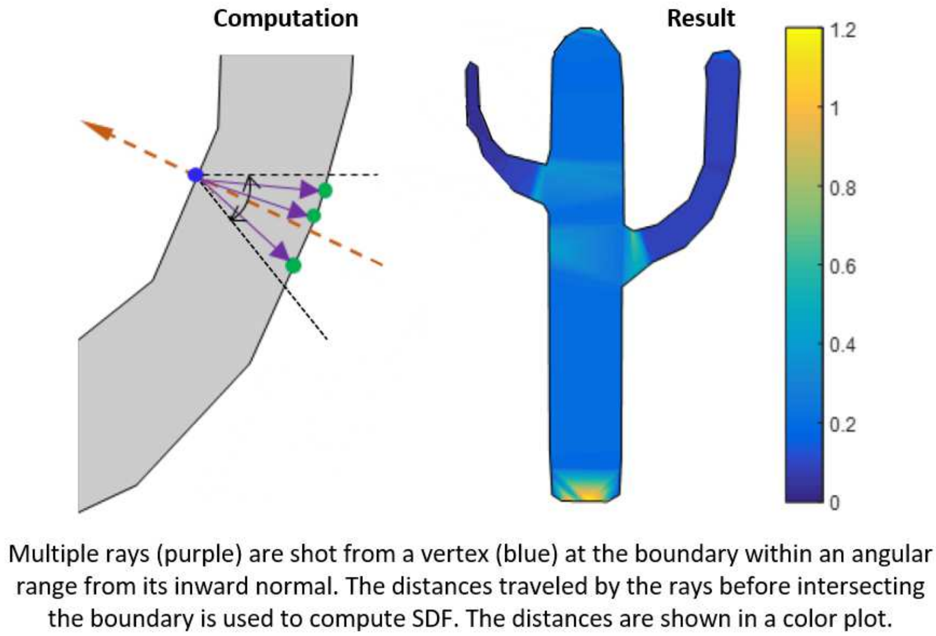


Figure 2.7: SDF adpated from [JR19].

Interior Angles

The interior angle at vertices can be used to determine sharp corners. Sharp corners can cause a problem with the printing process because of inappropriate fusion or stress concentration. By checking the internal angle between two edges not printable corners can be recognized. All vertices with an interval (Jaiswal and Rai [JR19] uses an angle of 150 to 210 degree) are considered as soft corners and printable. The others are detected as critical sharp corners (see Figure 2.8).

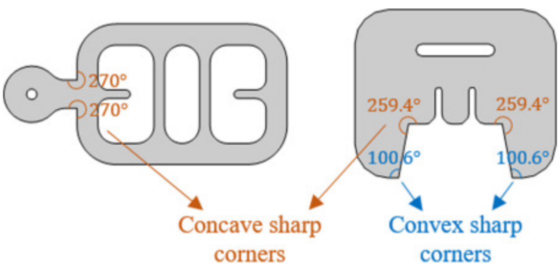


Figure 2.8: Interior angle at vertices on the slice boundary to identify sharp corners [JR19]

Morphological Operations

Morphological operations can be used to identify small holes and intrusions as well as identify small features especially walls. These can be done with the opening and closing function illustrated in Figure 2.9. The opening operation is obtained by the erosion

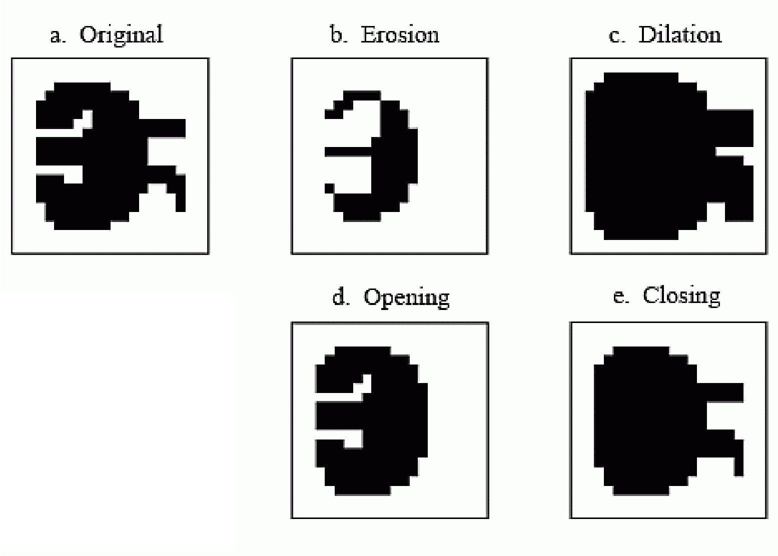


Figure 2.9: Morphological operations [mor19]

followed by the dilation of the slice by a structure element. The structure element represents the smallest printable feature in form of a disk that represents, for example, the nozzle diameter or laser beam width. After the opening operation, all features that are too small have disappeared. By forming differences to the original slices, these regions can be detected. The closing operation can be used to find small gaps and holes. It is obtained by the dilation followed by the erosion of the slice by a structure element. After the closing operation all gaps and holes with the size lesser then the structure element are closed. Again, by forming differences to the original slices, these critical regions can be detected. Nelaturi et al. [NKK15] uses the morphological opening operation to form a printable map of the object. Jaiswal and Rai [JR19] uses the morphological closing to detect small holes and intrusions in his manufacturability framework.

2.3.2 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 or with polygon meshes. Because the common mesh type in computer graphics and mesh processing is the triangle mesh,triangle-based analysis methods are presented. Another novel approach of object model analysis in three dimensions is the heat kernel signature.

Voxel-Based Analysis

The geometry of a solid model can be represented in different ways. Two possible representations are the surface or polygon-based representation and the voxel-based representation (see Figure 2.10). The voxel-based representation uses voxels to represent the shape of

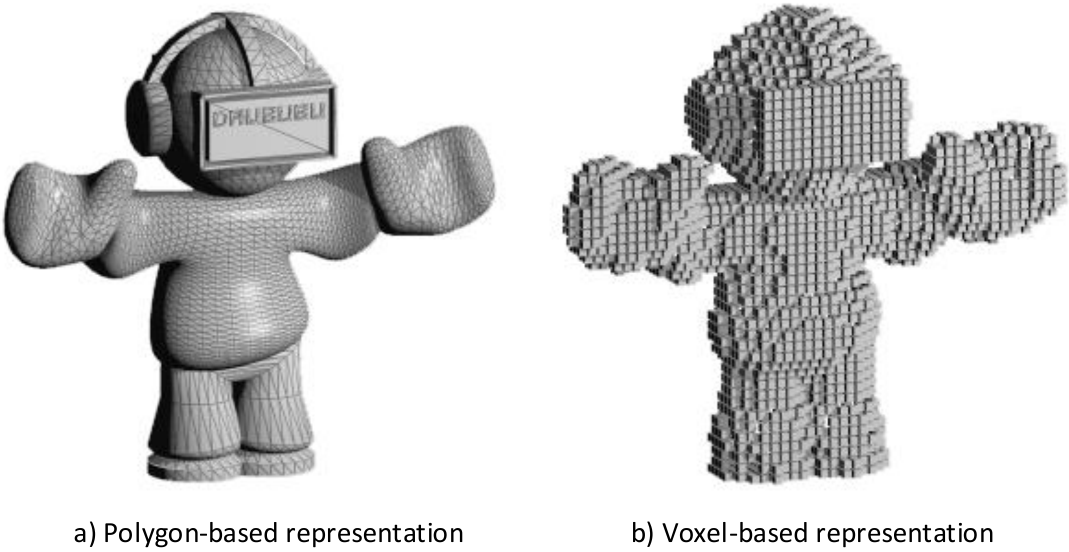


Figure 2.10: Geometry representation adapted from [Pic12]

the object. Voxels are cubes aligned to the cartesian coordinate system. These voxels are stacked to each other to represent the solid model. They are easily accessible through their x, y, and z indices. The advantage of this representation is that the voxels can be represented as a three-dimensional binary grid array. A value of *1* in the array means the voxel contains a part of the solid model and a value of *0* means it does not. The grid divided the bounding box of the polygonal mesh into voxels. This will be done with a ray casting method to form the voxels. The three-dimensional binary array can be stored on the Graphics Processing Unit (GPU) to allows a calculation in parallel. Furthermore, simple two-dimensional algebraic and Boolean operations can be used on the three-dimensional grid. In general, calculations on the part geometry with the voxel-based representation are easier than with the surface-based representation. However, there is also a disadvantage. The accuracy of the dimensions of the extracted features depends strongly on the voxelization resolution. Increasing the resolution leads to larger memory consumption.

Tedia and Williams [TW16] developed a framework for performing a manufacturability analysis of parts to be manufactured using a voxel-based representation schema, which provides feedback on unfeasible features, minimum feature size, support material, orientation and manufacturing time for different build orientations. A schematic diagram of the steps of their manufacturability tool is shown in Figure 2.11.

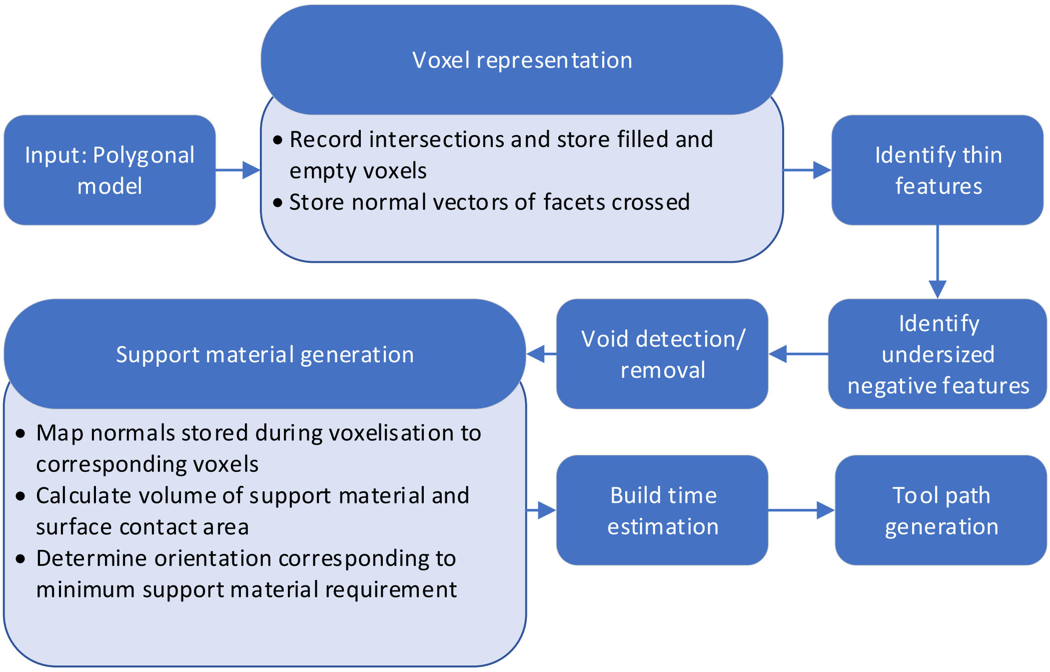


Figure 2.11: Flow chart of the voxel based analysis adapted from Tedia and Williams [TW16]

The voxel-based approach of Tedia and Williams [TW16] does not categorize features into categories like walls, holes or cylinders. Thus, for example, no explicit query can be made for hole radii. Voxels are counted to determine minimum feature sizes. Additionally, it is recognized if the features are negatives (notches, holes) by counting voxels between each set of even-to-odd intersection. Critical surface orientation and overhangs are investigated but not differentiated in their type (H or T-type). Cross-section areas and diameters are not mentioned in the approach but can be easily calculated using the voxel layer structure.

Triangle-based Analysis

Solid models can be represented uses polygon meshes. 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. The commonly used surface representation in additive manufacturing consists of triangles. The file format is called STereoLithographie or also Standard Triangulation/Tesselation Language (STL). The format contains three vertex points and surface normal for each triangle of the model. Figure 2.12 shows a sphere in STL format with information per triangle.

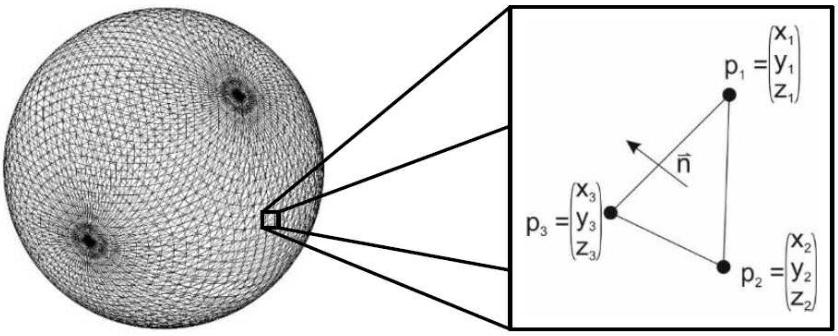


Figure 2.12: Sphere in STL format, showing the triangle information [TLWZ18]

By linking the triangles, the necessary geometric information, like distances, curvatures, and area sizes, can be calculated to verify the manufacturability features. Rudolph and Emmelmann [RE17] 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, for the calculation of some guidelines like the overhang, the orientation of the product and the build direction needs to be taken into account.
In this context, Tominski et al. [TLWZ18] proposed a software-based design check for AM concept. He builds on the results of Rudolph and Emmelmann [RE17] and extends their solutions with suggestions for missing guidelines. Under consideration of the building direction, the surface orientation is checked with the normal vectors of the triangles. They mentioned computing vertex normals as a weighted average of the neighboring triangle normals, which result in a more smooth normal field. To calculate the cross-section area it was suggested to slice the solid model in building direction in an appropriate step size depending on the model. The border of one slice can be used to calculate the area via the Gauss's area formula. Tominski et al. [TLWZ18] also describe the usage of the medial axis to calculate distances and radii. Further details of the use of the medial axis are presented in Section 2.6

Feature Extraction with Heat Kernel Signature

Heat Kernel Signature (HKS) is a pointwise shape descriptor developed in computer vision [SOG09]. HKS defines for each point in a shape a feature vector. The feature vector represents the point's local and global geometric properties. HKS can describe the shape of the surface and also the position of a point on a given domain. It is related to the Gaussian curvature on a surface and closely related to diffusion maps and diffusion distances.

HKS 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. To obtain the rate the heat diffusion equation has to be solved. The heat equation describes how the distribution of heat evolves over time in a solid medium. In order to solve the equation, the heat kernel is used. The kernel represents the evolution of heat in a region whose boundary is held fixed at a particular temperature. Shi et al. [SZB+18] 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 as shown in Figure 2.13(c) in red.

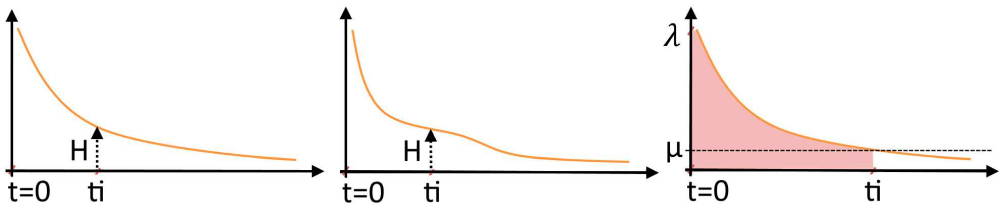


Figure 2.13: (a) Heat persistence at a typical point, (b) Heat persistence at Point with resistance areas, (c) Heat persistence value shown in red adapted from [SZB+18]

The vertices are clustered into different sets using the heat persistence value and a percentage similarity. These sets are separated through a multiscale clustering method. The tip cluster gets a special treatment to merge these clusters with others to form the whole feature. The merging is done based on similarity and inclusivity for similar subsets at incremented persistence similarity subsets. Identified subsets are assigned to faces they belong to, because of the geometric reasoning of vertices. That forms a set of faces that belongs to one feature. The process of the feature recognition process is shown in Figure 2.14.
With this method, only the features can be detected. However, lengths, angles, and 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. This information can be used to slice the object in these directions to calculate overhangs. The usage of the smallest distribution direction can be used to find the minimal feature size as well to calculate aspect ratios. Also, the main direction is used to calculate the angle between the object and the building platform the check if the object needs support.

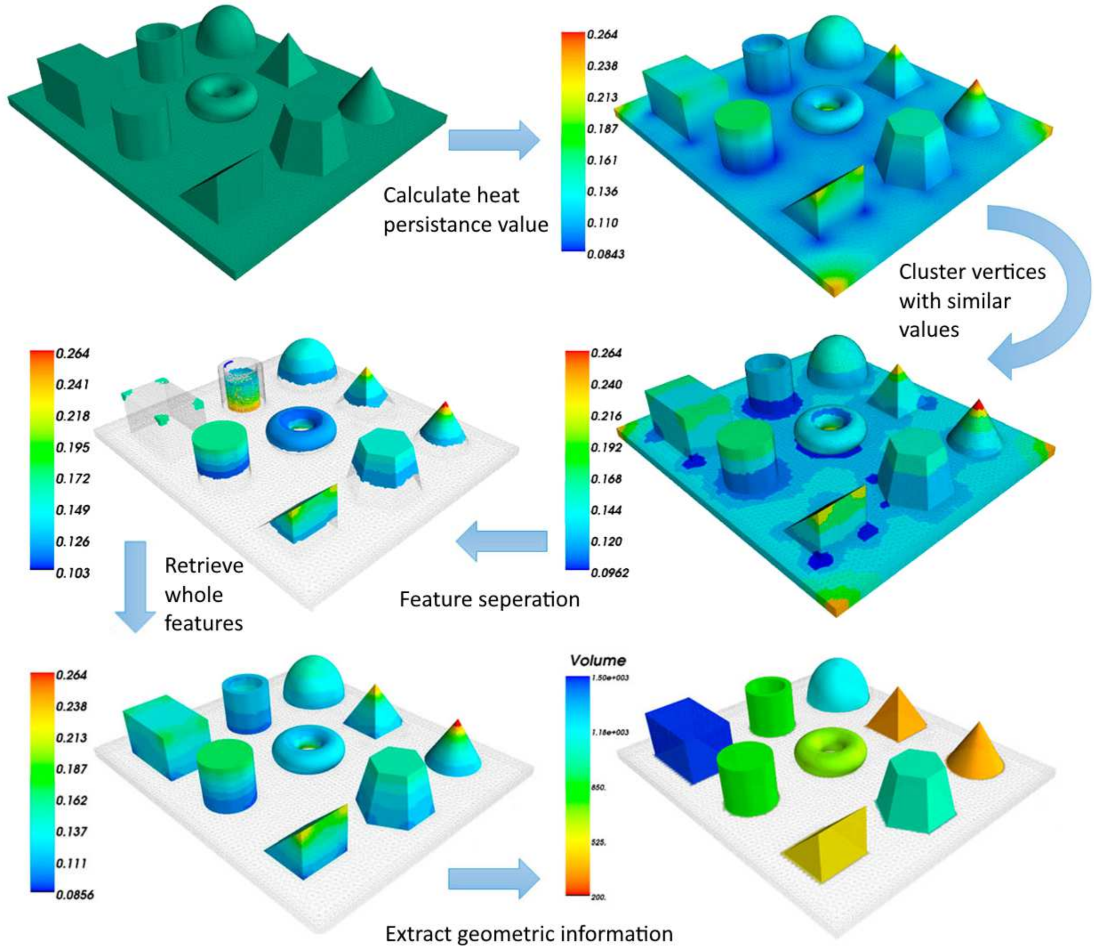


Figure 2.14: Flowchart of the HKS recognition procces adapted from [SZB+18]

With the HKS and SVD the approach by Shi et al. [SZB+18] supports the following guideline checks:

1. Minimum feature size
2. Overhangs
3. Surface orientation for self supporting structure
4. Minimum space between features
5. Maximum vertical aspect ratio

2.3.3 Overview Techniques for Guidelines

Table 2.2 compares the different methods for checking various guidelines in the manufacturability analysis papers.

	Guidelines					
Paper	Wall	Gap	Hole	Pin	CSA CSAR	Overhang
Nelaturi et al. 2015	M	/	/	/	/	/
Tedia & Willimas 2016	V (mFs)	V (mFs)	V (mFs)	V (mFs)	V (Layer)	V (NV)
Rudolph & Emmelmann 2017	T	T	T	T	/	/
Tomminksi et al. 2018	MA	MA	MA	MA	Gauss Formula	T (NV)
Shi et al. 2018	HK+SVD: (mFs)	HK+SVD: (mFs)	HK+SVD: (mFs)	HK+SVD: (mFs)	HK+SVD (feature)	HK+SVD (NV)
Jaiswal & Rai 2019	SD	M	M	SD	/	/

CSA	Cross-section area
CSAR	Cross-section aspect ratio
M	Morphological operation
V	Voxel-based
T	Triangle-based
SD	Shape diameter function
MA	Medial axis
HK	Heat kernel
mFs	Minimum feature size
NV	Normal vector

Table 2.2: Comparison of manufacturability analysis approaches and their guideline checks.

There are several different approaches to review guidelines using different methods. This table contains the most important guidelines, except the curvature, which is not dealt with within any of the approaches. Tedia and Willimas [TW16] as well as Shi et al. [SZB+18] present methods in which they can theoretically test all six guidelines. Their concept is based on checking the minimal feature size. However, there is no concrete distinction between the guideline categories that make the application of the methodology more difficult. There are other requirement values for the different features like walls, gaps, etc. The approach of Tedia and Williams [TW16] allows finding the voxels which are smaller than the minimum feature size by the voxel structure of the object. Another algorithm would be necessary to perform the feature categorization. Shi et al. [TLWZ18] extracts the features within an object using the HKS and then calculates the minimum feature size using the SVD. The extracted features would have to be categorized again. In general, the algorithms in Table 2.2 usually only check the minimum feature size and do not determine the shape. Exceptions are morphological operations, which define the found structure based on the structure elements used. Furthermore, the triangle-based algorithms of Rudolph and Emmelmann [RE17] aim the recognition of the feature structure.

2.4 Unsolved Problems and Proposed Solution

This work aims to design a prototype overall system for manufacturability analysis in AM. It should be an overall system that makes it possible to integrate all different AM processes. Thus, the definitions of all AM domains are specifically abstract. However, the work focuses specifically on the integration of manufacturability analysis in AM in the field of ceramic printing with the LCM process.
Several components are necessary for such a system. The basis is to define the guidelines that are necessary for the analysis. In the literature, some research has already been done on this, but also with open points. As can be seen in the guidelines from Kranz et al. [KHE15], 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. Guidelines are also dependent on the machine, manufacturing parameters and used material, which can be seen by the guidelines of the well researched AM process laser beam melting [LTM+18]. This dependency has not yet been modeled in a form that makes it possible to check it efficiently in an automatic manufacturability analysis software.
One way to bring the knowledge necessary for a manufacturability test, especially the mapping of guidelines, into a machine-readable, automatically processable format is an ontology. Further requirements for knowledge storage are the simple expandability of the system, as the technology is under development and new requirements are constantly being researched. Ontologies are flexible due to their structure as graphs and have already been researched in connection with the AM domain (see Section 2.2). Kim et al. [KRWK19] comprehensive work is a very good starting point for an ontology for an automatic manufacturability test. However, concerning this work goal, the approach still requires some improvements and changes. They stated to build a general manufacturability ontology but has also the focus on a specific process, the powder bed fusion. Furthermore, especially in ceramic printing post-processing and optimization of the process is an important part of the printing process. Kim et al. [KRWK19] also maps the geometry of the object in the ontology to check it for manufacturability with the query language SQWRL. In their work, they describe that the queries for the manufacturability are complex. Also, the geometry must be available in its defined format to carry out the test. You want to check geometry afterward, where all feature parameters are no longer known. However, the software is required to extract the necessary features. At this

point, it would then be possible to load the extracted data into the ontology and solve the manufacturability in the ontology with the complex SQWRL queries. To be able to construct a this long and complicated queries the information structure of the DFAM ontology must be understood. Another approach would be to use the ontology only as a knowledge base and let the software do the checks. For this work the second approach is chosen and the ontology is rebuild to provide a simple interface to the automated manufacturability analysis software.

Furthermore, 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. Jee et al. [JW17] defines a good concept with the formal design rules, which is also partly adopted by Kim et al. [KRWK19]. this work builds more on Jee et al [JW17] concept and again reflects the consequences, like manufacturable, self-supported, resolvable, survivable (cleaning by material jetting), in the ontology. Furthermore, a new simple guideline structure is defined in the ontology, which makes it possible to assemble guidelines in a modular way.

As mentioned above this works approach uses software to automatically check the guidelines from the ontology. Therefore feature extraction algorithms are needed. Section 2.3 has shown that there are already some papers that have dealt with a manufacturability system. There are several approaches to different guidelines. Section 2.3.3 states an overview of their approaches. However, some of them are only concepts and have not been checked for applicability. Besides, the checks only include 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. Overhangs come in various forms. These can be supported on one side. In technical terminology, this type is usually called T-type or two-sided supported which would form a bridge. A bridge is also called H-type. These and other small differences are not addressed in the existing approaches. This work is intended to contribute a general part to the manufacturability analysis in AM. However, it focuses especially on the manufacturability in ceramic printing with the LCM process. This procedure is subject to its guidelines, most of it is consistent with traditional guidelines. A significant difference with the LCM procedure to the other procedures exists with the requirement of the curvature. The method cannot print right angles, especially radii of curvature below a certain radius are not possible, because otherwise, that can result in cracks at this point. The curvature has not been addressed in any of the previous approaches. As the analysis of the state of the art guidelines has shown, there are requirements for the inner radii. The construction of the object in a triangle mesh complicates the exact calculation of the curvature. Especially CAD manufactured objects do not show an even triangulation.

This works manufacturability analysis algorithms are based on the triangle-based approach by Tomminksi et al. [TLWZ18] and extend its basic concept to include the above points. A further open point is that the state of the art approaches concentrates on simple geometries and more complex geometries are not covered. This work will not cover this aspect either. Together with the ontology and guidelines from the LCM process, this work forms a prototypical implementation of an AMMAS.

CHAPTER 3

Additive Manufacturing Manufacturability Analysis System

This work provides the basis for a cloud-based AMMAS. Cloud-Based design and manufacturing refers to a service-oriented networked product development model in which consumers can configure, select and utilize customized product realization services [WRWS15]. This section describes the architecture of the cloud-based AMMAS with a focus on manufacturability analysis. Furthermore, it would be possible to map the entire production chain in the cloud. This means that the object models themselves can be processed with cloud-based CAD systems. However, this is not included in the AMMAS. This architecture 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.

3.1 Overview of the Manufacturability System

The system integrates the following key components (see Figure 3.1): Mesh analyzer, ontology, and cloud platform that incorporates an ordering form and a mesh visualizer for the user. The whole system is connected with the cloud platform to ensure a service-oriented cloud-based solution.
The cloud platform provides the customer with the interaction interface to the framework. First, the desired product is entered usually in the STL format to examine the mesh in regard to possible imperfections. 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. 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 manufacturability check delivers, as a result, a triangulated mesh with annotations of the critical areas. The critical features are visualized in different colors. The different colors should represent different consequences. For instance Figure 3.1 shows an object with a hole in red, which has a radius smaller than the tolerable radius according to the guidelines.

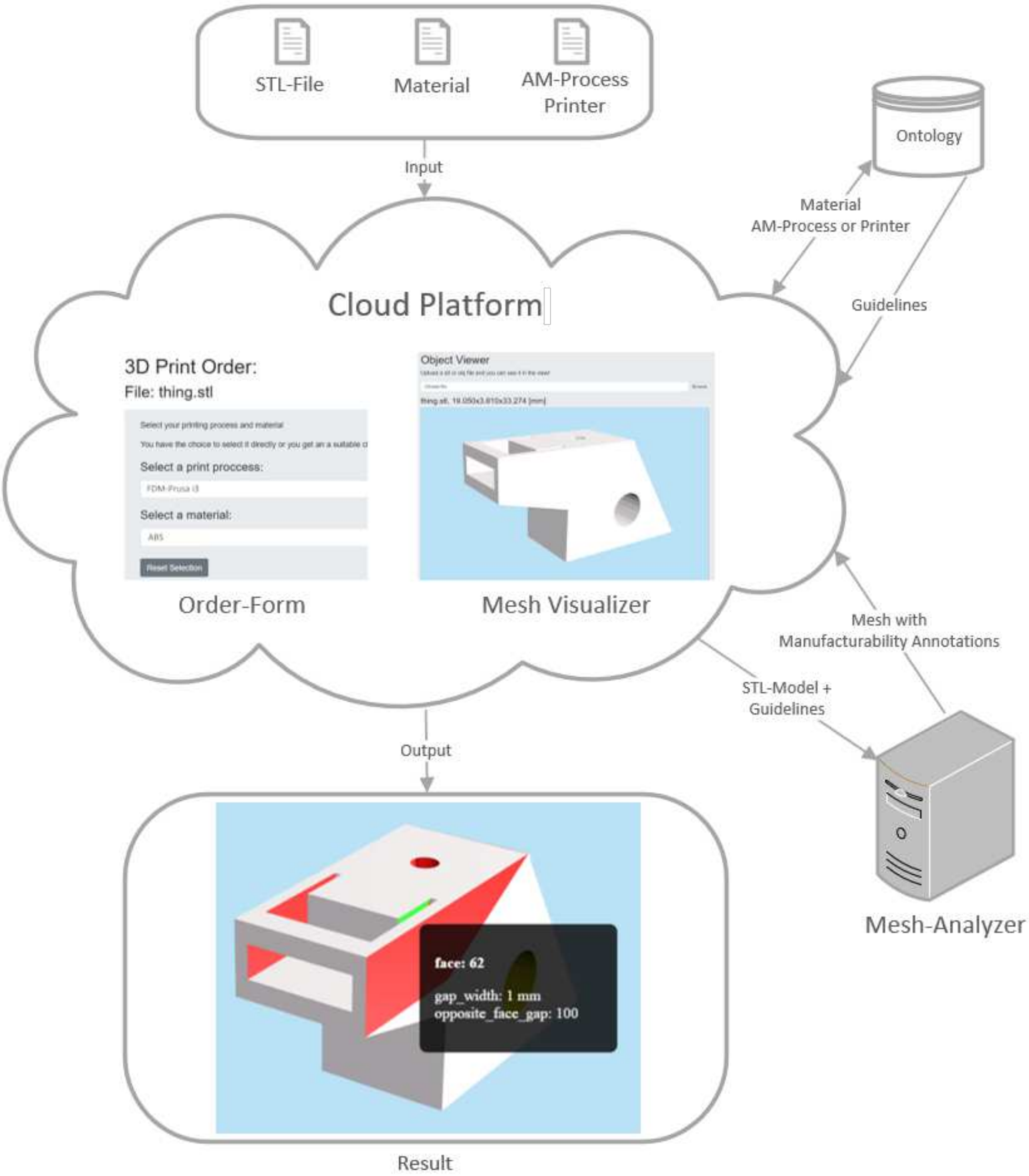


Figure 3.1: Architecture of the framework for manufacturability analysis

The procedure of the manufacturability check is illustrated in the flow diagram in Figure 3.2. To perform a manufacturability check, you must first extract the guidelines required for this component. The guidelines are represented in the AMMAS in guideline sets in the ontology (detailed description see Section 3.3.1). For the exact determination of the guideline set, data of the used AM process as well as printer, manufacturing parameters, and used material are necessary. However, ontology also offers abstract guidelines, where it is not yet necessary to know the printer type and its manufacturing parameters to be able to make a rough pre-check. The extracted guideline set is the input for the mesh analyzer. The analyzer applies the necessary algorithms for the guidelines to the object to perform the analysis. 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.

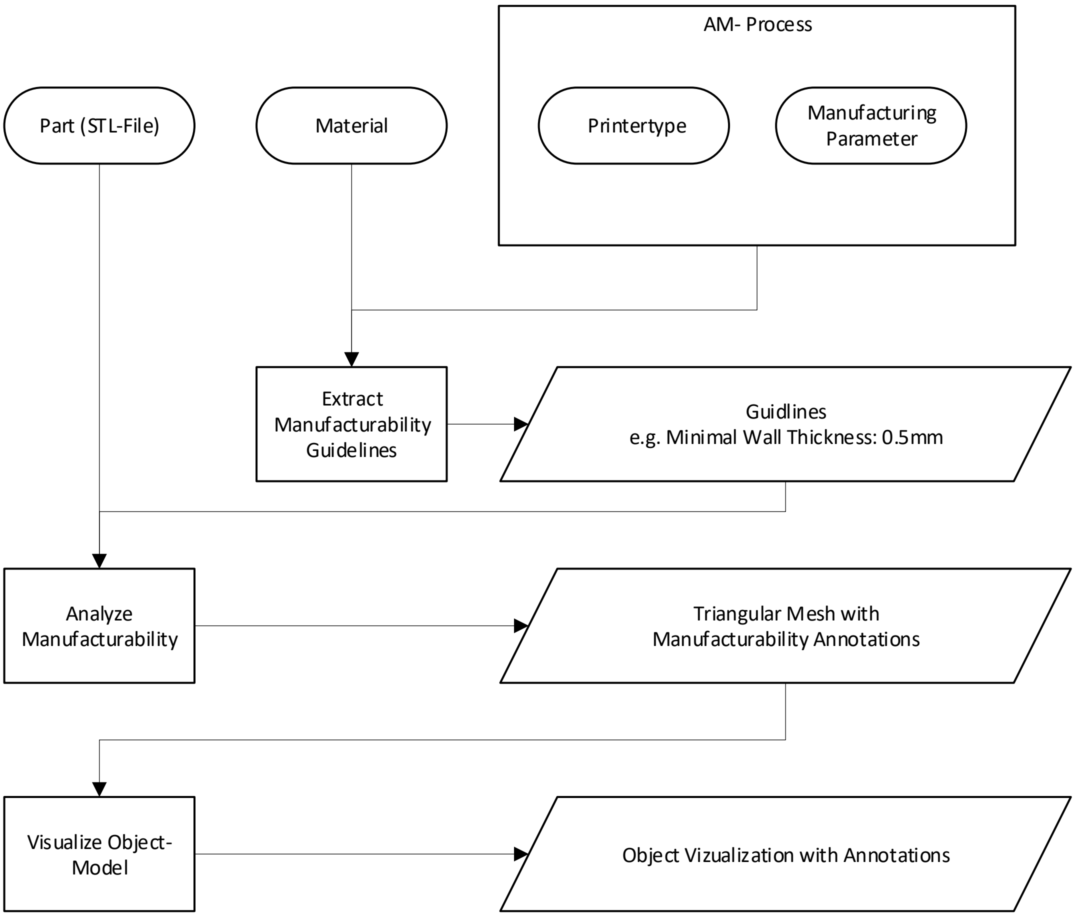


Figure 3.2: Flow diagram depicting the procedure of the manufacturability check

3.2 Mesh Analyzer

The quality of the printed part depends on a relationship between 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 dimension of the walls and holes in the part, since too thin walls could lead to thermal dissipation and cause different defects, such as un-melted powder inclusions, internal voids, cracks and shape irregularities [SZB+18]. Generally, the design process should be done in several loops considering the final function of the specific part but also performing manufacturability analysis involving all relevant aspects. According to the state of the art methodologies exist different approaches for the manufacturability analysis.
In this work, the manufacturability check is performed with the mesh analyzer. The analysis mainly operates in three-dimensional space. The only exceptions are the cross-sectional areas, where a layer (slice) is examined. The mesh analyzer uses the triangle representation of the objects to perform a triangle-based analysis as described in Section 2.3.2. This work is based on the approach presented by Rudolph and Emmelmann [RE17], which uses the triangulated surface geometry in STL format for automatic analysis and assessment of a part's geometry. The format contains three vertex points and a normal for each triangle of the model (see Figure 3.3). By linking the triangles, the necessary geometric information, like distances, radii, curvatures, and area sizes, can be calculated to verify the manufacturability features. Rudolph and Emmelmann's approach is rather

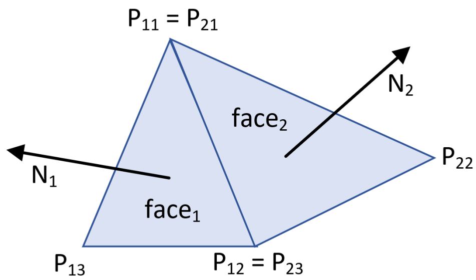


Figure 3.3: STL format: Triangle connection with corner points and surface normal.

focused on relatively simple part geometries and only implements four common guidelines wall thickness, gap width, hole radius, and pin radius. But 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, as presented in the work of Lammer et al. [LTM+18]. Besides, for the calculation of some guidelines like the overhang, the orientation of the product and the build direction needs to be taken into account. In this context, Tominski et al. [TLWZ18] presents ideas on how to consider guidelines that take the building direction into account. They describe using the normal vector to check the surface orientation and check overhang angles. Furthermore, they present the idea to calculate the cross-section area with the Gauss's area formula on one slice of the object. With this basis, the algorithms were developed and adapted to the requirements of the LCM process. Other algorithms for guideline checks that have not been considered so far, such as the curvature or differentiation of the overhang type, are also developed.

The following guidelines for the LCM process have been researched so far and are therefore also being examined:

- Wall thickness
- Gap width
- Hole radius
- Channel size
- Pin radius
- Pin ratio
- Overhang without support structure (H, T-type)
- (Curvature) inner,-outer radius

The following sections describe the guidelines, provide a brief overview of the algorithm and a detailed description with illustrations. First, in Section 3.2.1, the general topic of the runtime is discussed and a solution by the usage of a space-partitioning data structure k-dimensional tree (k-d tree) is presented.

3.2.1 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. This leads to the fact that the number of triangles of an object is very high. Depending on the triangulation resolution, the number of 1000 triangles is sufficient for small structures with few curves to half a million of triangles for more complex structures. This leads to an immensely high runtime for algorithms that compare each triangle with the others. A detailed explanation of the runtime is given in Section 4.4. The naive algorithm compares each triangle with each other in space that corresponds to a runtime of $O(n^2)$. This exponential growth is not practicable for object with a high resolution. One solution for this problem is to subdivide the space. It is not necessary to compare a triangle with a triangle at the other end of the object if you only want to examine the minimum wall thickness of two millimeter and the object has a width and length of 100 millimeter. The space-partitioning data structure k-d tree is one possibility. It is also used in computer graphics for faster rendering using ray tracing. The k-d tree is a binary tree that at each level of the tree divides the space along an axis with a hyperplane into two parts. K-d stands for k-dimensional and therefore it can work with k-dimensional data. For this application the k-d tree is used to divide the three dimensional space. The sequence of the subdivisions is freely selectable. The subdivision is performed according to the X, Y, Z axis (see Figure 3.4).

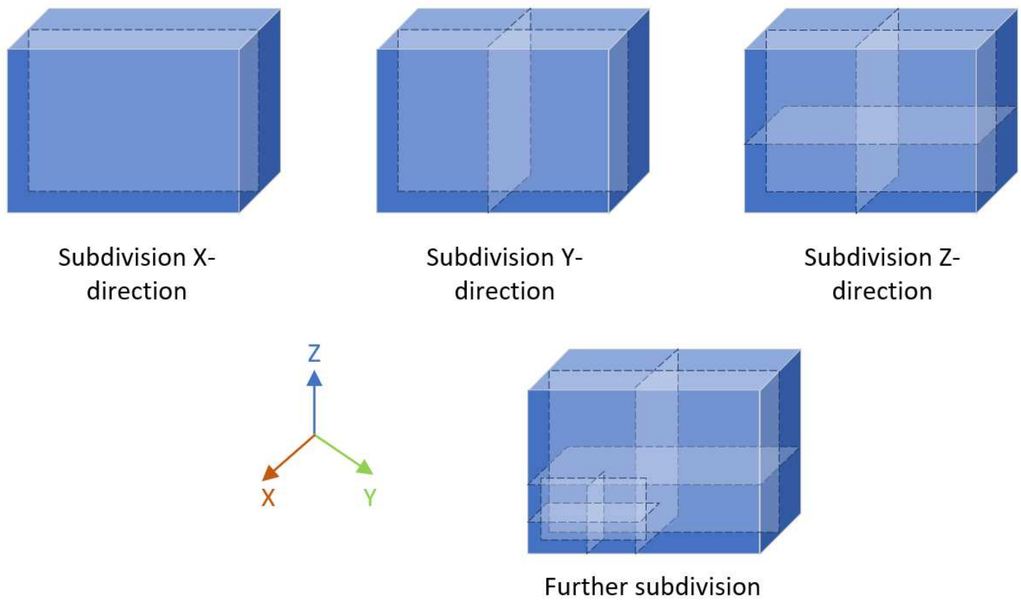


Figure 3.4: k-d tree construction

The construction of the 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 tested. This is done by an intersection test with the enlarged triangle bounding box and the two bounding boxes. The bounding box of the triangle is increased by the distance of the minimum wall thickness. The bounding box has to be enlarged because a triangle can be located at the edge of the bounding box of one node and so no test with the other node would be done. If the intersection test is successful the triangle will be added to the node. This is repeated recursively up to a termination condition. The chosen termination condition 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 $O(n * log(n))$. In the leaf node, the triangle is then only examined with the remaining triangles within the leaf node.

3.2.2 Wall thickness and gap width

An important requirement for the printing process is the minimum wall thickness or, in the case of ceramic printing with the LCM process, also the maximum wall thickness. Wall thickness is the distance between two surfaces. If there is a solid between the two surfaces, it is a wall. If there is only air between the two faces, it is a gap.

Algorithm 3.1: Wall thickness and gap width

Input: Triangle mesh
Output: Matrix of triangle to triangle distances below the threshold and opposite faces

1 Detect opposite faces;
2 Examine triangle normals to categorize in wall or gap;
3 Calculate triangle to triangle distance;

The naive algorithm checks each normal with each other normal. The implementation with the k-d tree uses a subset of triangles. Faces with an obtuse angle to each other (angle > 90 degrees, dot product < 0) are considered opposite (see Figure 3.5).

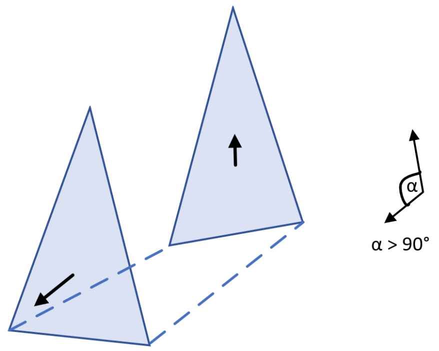


Figure 3.5: Opposite triangle faces: Greater than 90 degree

Afterward, it is checked whether the two normals stand to each other or opposite to each other. The check is done by calculating the distance between the two triangles. The normal 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 normal 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 (see Figure 3.6). Finally, the minimum distance between the two triangles is calculated using the algorithms by Ericson [Eri05].

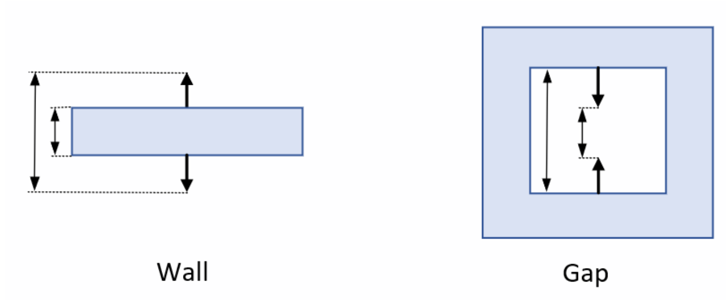


Figure 3.6: Wall and gap difference with distance measure

3.2.3 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. Pins and holes can be manufactured smaller due to their shape. Looking at the surface of a pin and a hole they are identical objects with different normals. Therefore, circular structures are searched, which are then differentiated according to the normal. With pins, the normals point away from each other and with holes to each other.

Algorithm 3.2: Hole and pin detection and radius calculation

Input: Triangle mesh
Output: Holes and pins and their radii

1 Detect circles;
2 Find two farthest away faces;
3 Calculate triangle to triangle distance;

First, a contiguous structure is searched that forms a circle (see Figure 3.7). Here the normals 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 normal. Therefore adjacent faces with the same surface normal (angle $= 0$ degree, dot product $= 0$) are included as well. The acute angle between the triangles is buffered and compared with the next possible adjacent triangles. If the 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 [Eri05]. The distinction between pin and hole is analog to the distinction between wall thickness and gaps in Section 3.2.2.

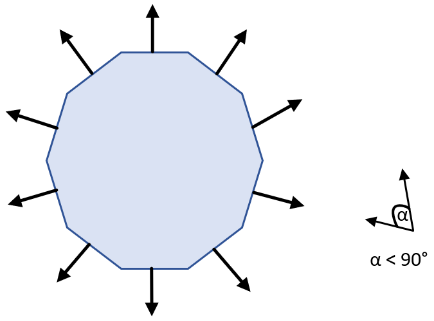


Figure 3.7: Triangulated circle with normals

Pin Ratio

The extracted circular structures that form a pin are also used to calculate the pin ratio. This is a special guideline in the LCM process.

Algorithm 3.3: Pin ratio

Input: Pins
Output: Pin ratio
1 Calculate bounding box;
2 Calculate aspect ratio;

The aspect ratio can easily be calculated from the bounding box of the pin.

Channel Size

Another guideline in the LCM process is the verification of the channel size. A channel is a continuous hole. This means that by looking through the channel, it can be seen through the object. The check is done by analyzing the holes.

Algorithm 3.4: Channel size

Input: Holes
Output: Channel size
1 Find one face at the top of the structure and one at the bottom;
2 Compare normals;

From the extracting hole structure, two triangle faces are searched at the respective ends. The normals are then compared to each other (see Figure 3.8). If the two faces point in the same direction, it's a hole, if they point away from each other, it is a channel.

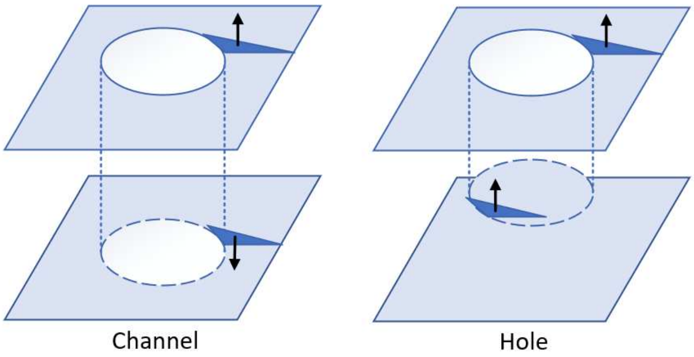


Figure 3.8: Difference between a channel and a hole with adjacent normals

3.2.4 Overhang and Surface Orientation

Overhangs are structures that are at a special angle to the building direction. They are hanging in the air. The angle between the surface of the overhang structure and the building direction determines whether it is an overhang structure or not. It is important to recognize overhang structures because they need a support structure at a certain length. If a structure needs support material, it must also be removed at this point. The removal of the support structure leads to reduced surface quality. Usually, a surface requires a support structure from an angle of 25 degrees. However, this also depends on the length of the overhang. There are also two distinctions, whether the overhang structure is unilateral attached or multilateral attached. With unilateral attached overhang structures one end of the structure is floating in the air. In the technical language is also called T-type, because of the T shape. Bilateral overhang structures are supported on both sides and form a bridge. Furthermore, this form is also called H-type in technical language. These two forms have different length requirements which can be printed without a support structure. The possible length of bilateral attached overhang structures without support structure is longer due to the greater stability.

Algorithm 3.5: Overhang and surface orientation

```
Input: Triangle mesh
Output: Overhangangle, -type, and -length
Detect surfaces with critical angles;
Assemble overhang structure;
Detect start and end face of the overhang structure;
/* Length calculation                                    */
Find a face that is attached;
Establish plane equation;
Determination of the overhang type (H or T);
Calculate length;
```

First, all surfaces are examined concerning the angle of the building direction. Surfaces that have a smaller angle than the critical angle will be further investigated. The critical angle is the minimum angle necessary to print the segment without a support structure. Critical surfaces are now subdivided into connected components. The beginning and the end of the structure are detected in each component. Then the calculation of the length between the two faces and the identification of the overhang type begins.
First, the attached face is searched. If there is a T-type overhang there is only one side with attached faces, if it is an H-type there exist two or more sides with attached faces. 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 Figur 3.9). The maximum refers to that the minimum length per point of the other side is calculated and the maximum value is taken. Ericson's algorithms [Eri05] were used for this purpose. The distinction between H and T-type is based on the orientation of the adjacent surfaces of the start and end-faces. The H-type overhang structure is identified by the face ends that are attached to faces that are in the opposite direction to the building direction (see Figure 3.10). If only the start or end-face has an adjacent face that pointing in the opposite building direction it is a T-type. If no face points against the building direction, the object stands the ground.

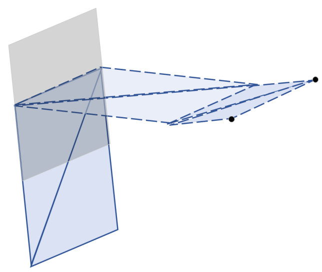


Figure 3.9: Overhang length calculation: Maximal (concerning two points) minimal distance between plane (gray) and the two points (black).

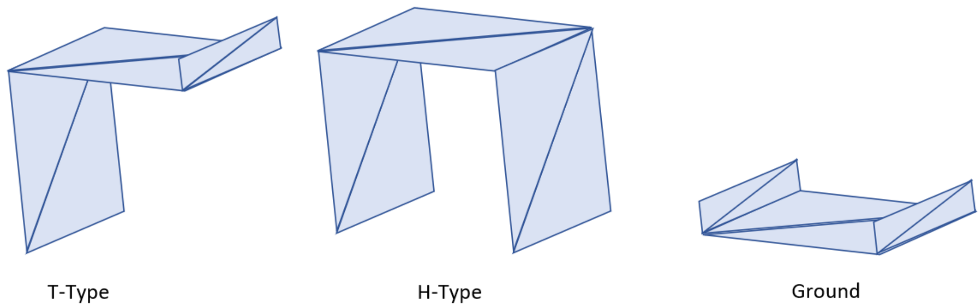


Figure 3.10: Overhang types

3.2.5 Curvature (inner, outer radius)

Requirements on the inner and outer radii of structures exist in several printing processes, as well as in laser beam melting from Lammers et al. [LTM+18]. In general, the radii are also referred to as the curvature of the object, which is of particular importance in the LCM method. The method cannot print right angles, especially radii of curvature below a certain radius are not possible, because otherwise, that can result in cracks at this point. Therefore, the curvature between each triangular surface is examined.

Algorithm 3.6: Curvature

Input: Triangle mesh

Output: Maximum Curvature between face adjacency's

```
Project three triangle vertices on a plane;
Calculate curvature with formula by Zhou et al. [ZYYW11];
```

The curvature is determined by the method of Zhou et al. [ZYYW11]. This method requires three points and calculates the radius of a circle through these points. Therefore two adjacent triangles are taken and their not common vertex and a common vertex are projected on one plane (see Figure 3.11). These three points are then used to calculate the curvature or radius.

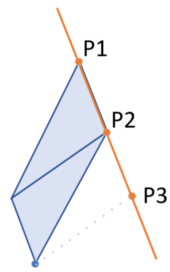


Figure 3.11: Curvature calculation with three points by Zhou et al. [ZYYW11]

3.2.6 Cross section area/ratio

Maximum cross-sectional areas are to be considered when it comes to the stability of the object. The area of slices is interesting or the ratio of the area in building direction. Even if 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 object is sliced into layers in 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.

Algorithm 3.7: 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;

3.3 Ontology

An ontology is chosen for knowledge modeling in this work because of the evolving nature of AM technology [KRWK19]. Compared to relational databases, ontologies make it easier to add new knowledge or modify legacy data. The AM technology at the moment is constantly in development. New processes are introduced, new material choices are available. To be able to map this flood of information, it is necessary that the knowledge base can be easily updated with new information. They are also more pertinent to a tutoring system and provide richer models [DR17]. 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. The knowledge base is modeled in the Web Ontology Language (OWL). OWL represents the rapidly evolving knowledge with a formal structure. The terminology and relations the OWL language helps users as machines to search and reuse information [EKG+15]. That makes OWL a human and machine-readable language.
The work's ontology formalizes existing knowledge about the AM domain. The goal for this work is that the ontology represents the knowledge about the manufacturability of parts geometry. This work especially focus on representing the requirements for the manufacturability in ceramic printing with LCM technology [FGM+12].
In the development of ontologies, it is important to consider existing work and to build on or reuse it. As mentioned in the state of the art Section 2.2 there exist a couple of ontologies in the field of AM. The ontology by Kim et al. [KRWK19] includes most of the state of the art until the year 2018 and is a great starting point. Kim et al. [KRWK19] also uses the ontology to map the parts geometry and its features for automatic reasoning on manufacturability with the ontology. The works model does not include the geometry of the part in the ontology. The ontology is used to extract the knowledge for the manufacturability which is then verified with the mesh analyzer. It would also be possible to apply Kim et al. [KRWK19] knowledge model and extract the features from the mesh analyzer and feed it into the ontology. Afterward, the manufacturability test can be performed in the ontology. This work is extendable to integrate the idea of Kim et al. [KRWK19] but in this prototype, the manufacturability tests are performed with the mesh analyzer.
However, it still contains geometric properties and geometric features. These are not used to describe the geometry of the part, but to be able to describe guidelines modularly, as in Jee et al.[JW17] approach. They utilize primitives and modules as described in

the state of the art Section 2.1 to form design rules with consequences. This concept is implemented with object properties in the ontology. Moreover, several sources have stated that the guidelines depend on the material, machine, and machine parameter to define a guideline precise [KHE15, TLWZ18]. This dependence is emphasized in the Ontology by guidelines sets, which define the dependencies of guidelines (see Section 3.3.1). To ensure a uniform terminology the ontology uses the terminology of the ASTM 52900 standard [Int15]. Ontologies are designed to build a general knowledge base. The idea of the ontology is to make it easy extendable to a general manufacturing ontology. In cooperation with another project which focuses on industrial robotics, this work consider the Rosetta ontology [ros]. Rosetta focuses on industrial devices and robotic skills but it modeled the concepts in a general manufacturing manner. Based on the Rosette [ros] ontology and the ontology developed by Kim et al.[KRWK19] this works ontology has the following five main classes:

- Capability
- Parameter
- PhysicalObject
- Property
- Skill

Figure 3.12 presents an overview of the ontology. Further description of the classes follows in the next sections. The focus of the work is to represent the manufacturability of the LCM process. Consequently, the ontology contains all necessary parts like materials, guidelines, machines, and processes for ceramic printing. Additionally, are data from the widely used Fused Deposition Modeling (FDM) process included as well as guidelines from other technologies [BR17].

3.3.1 Guideline set

Guidelines depend on many factors such as process, machine, manufacturing parameters and the material used. To be able to model this dependency, the ontology uses guideline sets as illustrated in Figure 3.13. The guidelines set defines the environment and the geometry guidelines applicable to the environment.

Table 3.1 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 via the "..." characters.

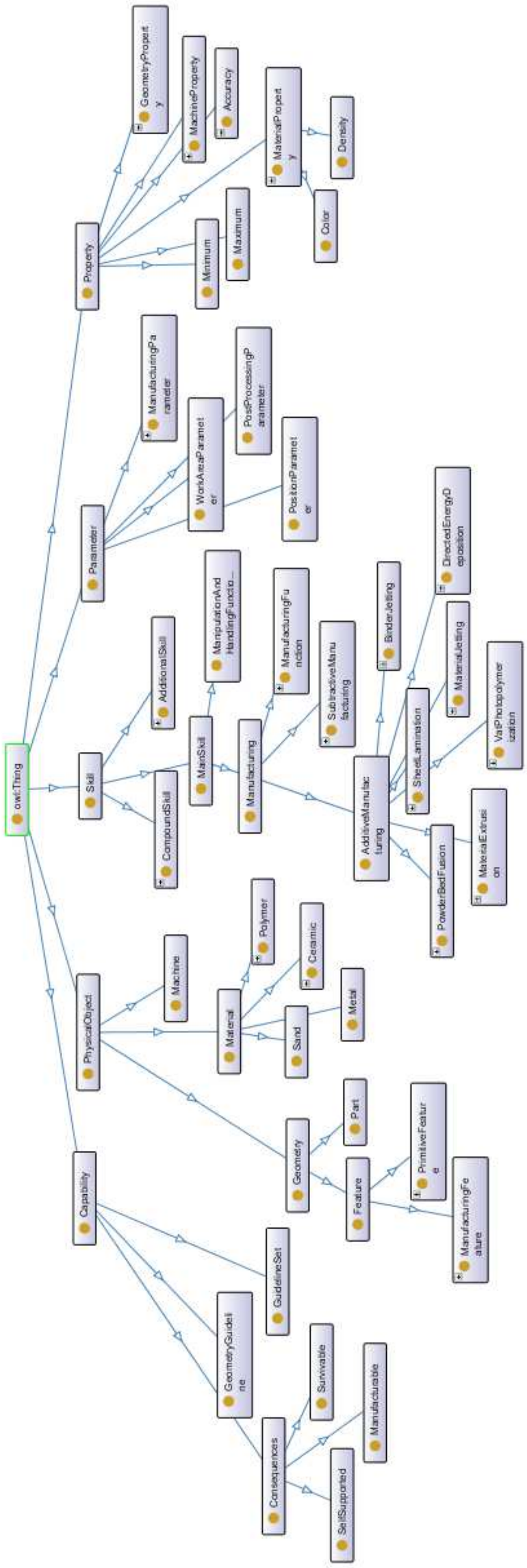


Figure 3.12: Overview ontology

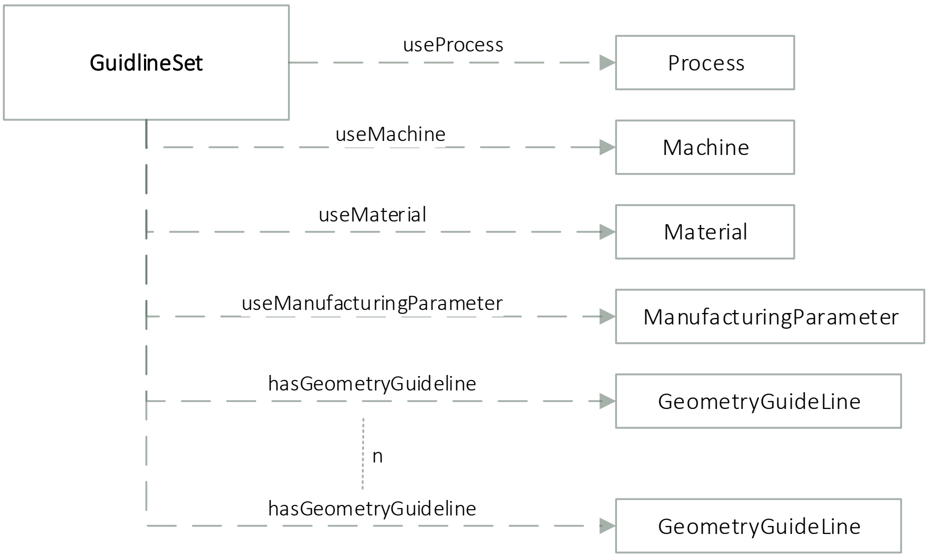


Figure 3.13: Guideline set: Relation to other classes

3.3.2 Guideline

Guidelines are available in various forms such as, only descriptive text[MW15], text with illustration [KHE15, TLWZ18, AZ15], 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 some naming conventions are based on the design rule approach by Jee et al. [JW17]. The more precisely the guidelines are defined in a machine-readable manner, the less development work is required to process the guidelines for a manufacturability system. The composition of a guideline from the various components is illustrated in Figure 3.14. An example guideline should make the concept clearer. A T-type (uniliteral attached) overhang below the maximum length of 2.1 millimeters is self-supporting. Divided into the individual parts means

Guideline set	**Object property**	**Object**
GuidelineSet_LCM	useMaterial	Alumina
GuidelineSet_LCM	useProcess	LCM
GuidelineSet_LCM	useMachine	CeraFab7500
GuidelineSet_LCM	useManufacturingParameter	MP_CeraFab7500
GuidelineSet_LCM	hasGeometryGuidline	OverhangLengthTType
...	...	...
GuidelineSet_LCM	hasGeometryGuidline	AspectRatioMaxCylinder

Table 3.1: Resource Description Framework (RDF) representation of a *GuidelineSet*. (Each line in the table represents a RDF triple)

that the guideline is applied to an overhang construction with the condition that is unliteral attached. The guideline should check the maximum length of the overhang. If the requirements are fulfilled, the consequences occur. In this case, the overhang would then be self-supporting. Table 3.2 represents the example guideline also in RDF format. Each line in the table represents an RDF triple.

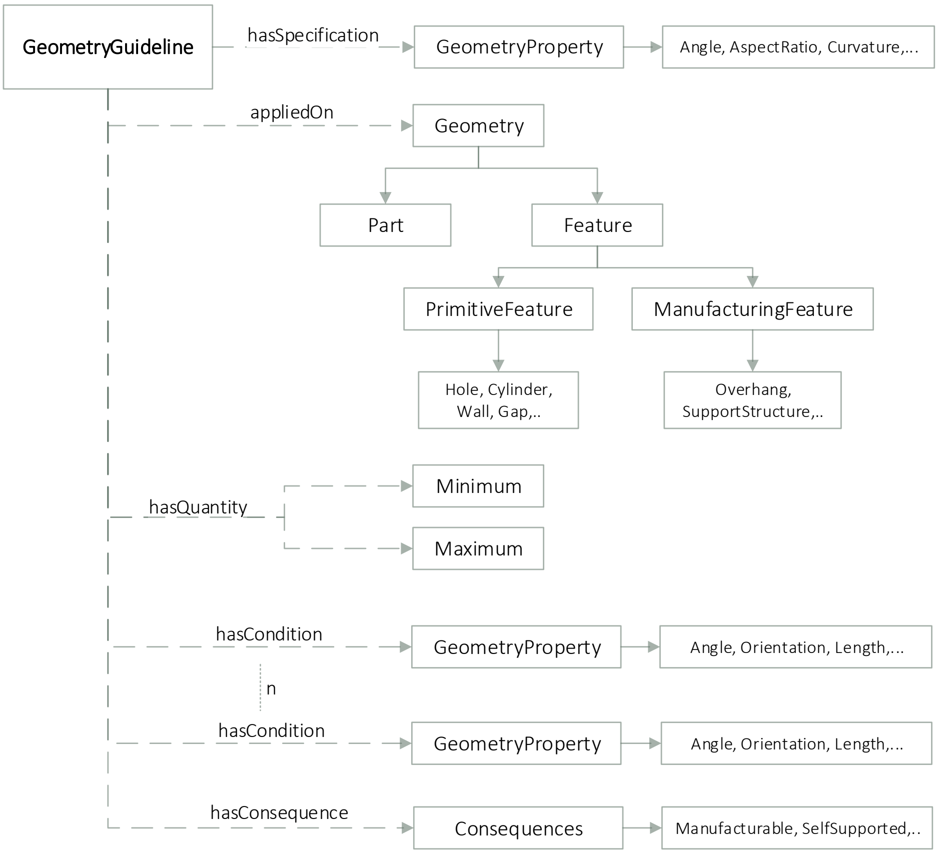


Figure 3.14: Geometry guideline: Relation to other classes

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

Table 3.2: RDF representation of a *Guideline*. (Each line in the table represents a RDF triple)

3.3.3 Capability

The *Capability* class represents the capabilities of things. In particular, process and machine capabilities are presented in this class. In AM processes some guidelines must be full filled to correctly manufacture the geometry of a part. These guidelines depend on various parameters. The representation of these dependencies builds the guideline set. 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 [MW15]. So the guideline consequence is that the feature is survivable during cleaning. Capabilities in contrast to skills are not executable. Figure 3.15 shows important geometry guidelines in AM processes. Especially the guidelines for the LCM process are included. In general, the most relevant guidelines are included [RE17, LTM^{+}18, BR17].

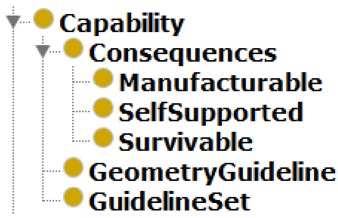


Figure 3.15: Ontology class *Capability*

3.3.4 Parameter

The *Parameter* class represents the parameters of things. Based on the Rosetta ontology [ros] the class is divided into *ManufacturingParameter*, *PositionParameter*, *PostProcessingParameter*, and *WorkAreaParameter* (see Figure 3.16). 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.

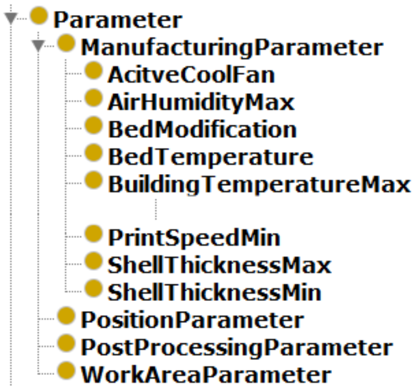


Figure 3.16: Ontology class *Parameter*

3.3.5 PhysicalObject

The *PhysicalObject* class represents the world. These are physical objects that are tangible in contrast to the other classes which represent concepts. Currently, it only contains classes that are necessary for exemplary use in the field of AM. Therefore it contains the class *Geometry* to describe the geometry of the object in the form of a feature or as a whole part (see Figure 3.17 and 3.18). There it will be differentiated between primitive features [AZ15] and manufacturing features [KRWK19]. The printer which executes the AM process are modeled via the *Machine* class. The product to be manufactured can be produced from various materials. The class *Material* represents the different material categories they are usually used in AM processes. The focus here is on ceramics and a small overview in thermoplastic has also been integrated.

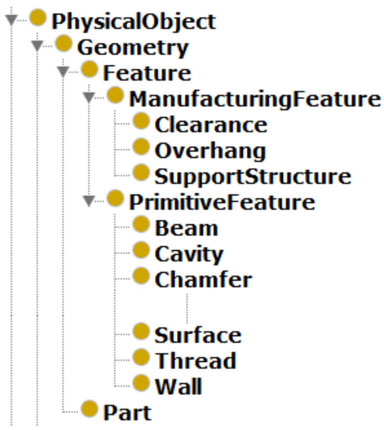


Figure 3.17: Ontology class *PhysicalObject* 1/2

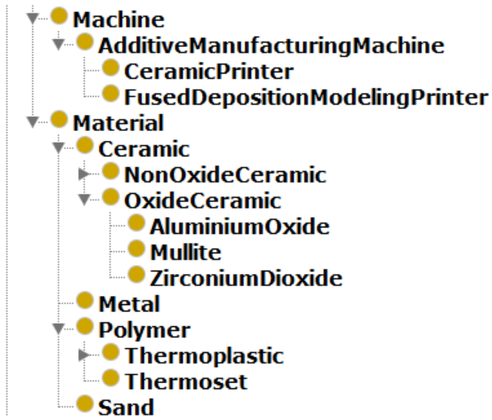


Figure 3.18: Ontology class *PhysicalObject* 2/2

3.3.6 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 Figure 3.19). Accuracy was modeled as a standalone class because it is not just machine precision. It defines the accuracy of something. This can be a process (the different AM-process have different accuracy [BR17]) or a machine.

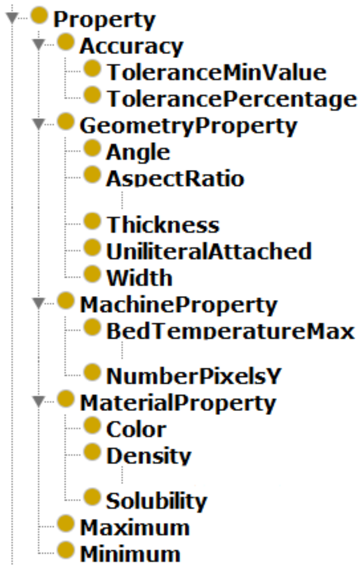


Figure 3.19: Ontology class *Property*

3.3.7 Skill

The *Skill* class represents the skills of things. Based on the Rosetta ontology [ros] the class is divided into *AdditionalSkill*, *CompoundSkill*, and *MainSkill.AdditionalSkill* describes skills like 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 52900 standard [Int15] AM is divided into seven categories (see Figure 3.20). Furthermore, processes belonging to the categories are also presented.

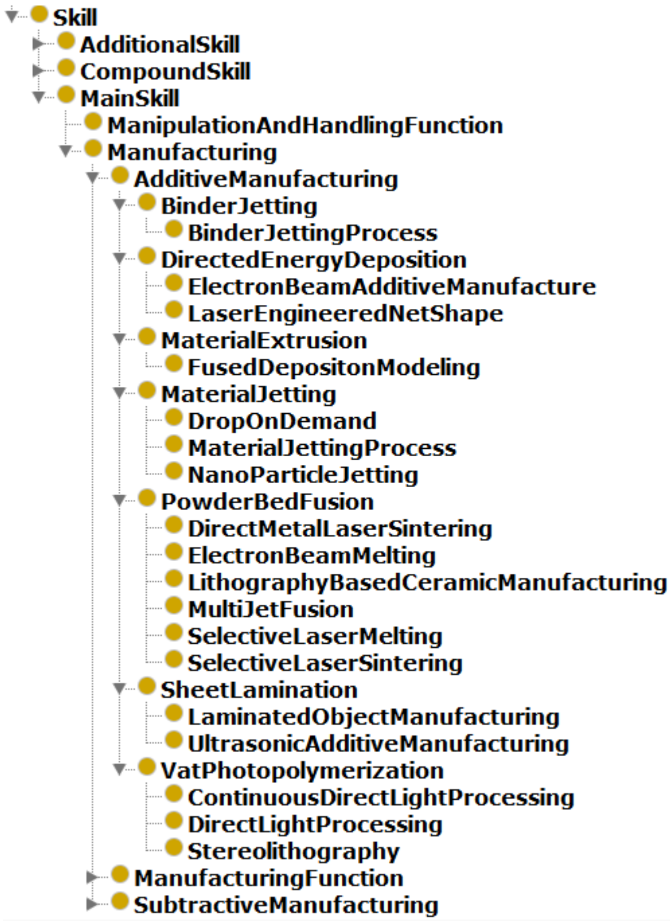


Figure 3.20: Ontology class *Skill*

3.3.8 Relation – Object Properties

To give the ontology even more semantics there are the concept of individuals, object properties and data properties [BvHH+04].Object properties link individuals to individuals. Datatype properties link individuals to data values. The relations for the guideline and guideline sets are already described in Section 3.3.2 and 3.3.1. Further relations are listed here to make general existence and usage queries possible. The used object properties are shown in Figure 3.21. *hasManufacturingParameter* assigns recommended parameter settings to machines, manufacturing process or materials. Based on material, appropriate parameters are recommended for manufacturing. *hasProperty* and his subcategories relate the properties to the individuals. *canUseMaterial* models which material can be used by which process. *applyProcess* links the machine with the manufacturing process.

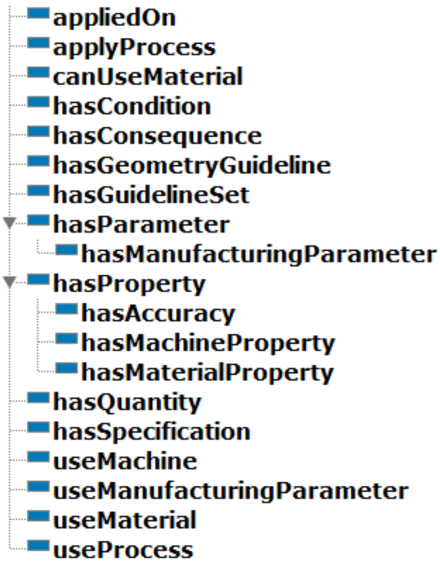


Figure 3.21: Ontology object properties

3.4 Realized System

This section describes the prototype of the AMMAS. As described in section 3.1 the AMMAS consists of three key components: Mesh analyzer, ontology, and cloud platform. The cloud platform is connected with the ontology and an external server on which the mesh analyzing service is allocated. The ontology is modeled using Protégé [pro] 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 the version 3.6.9 [pyt]. The core implementation is done with the scientific computing library NumPy in the version 1.15.0 [num]. The library PyMesh in version 0.2.1 [Zho] 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 PLY format (see Section 3.4.1). 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 (see Section 3.4.2).

3.4.1 Polygon File Format (PLY)

PLY is a file format for storing three-dimensional data that was originally designed for use with 3D scanners. The format is characterized by a simple description of individual objects as lists of polygons. These are defined as a list of vertexes followed by a list of faces. An advantage of this format is the possibility to define attributes for vertexes or faces. This feature has been exploited to add manufacturability annotations per face. An example of an object with manufacturability annotations is shown in Table 3.3.

```
ply                                      \\header begin
format ascii 1.0                         \\format version number
comment Generated by PyMesh              \\comments keyword specified
element vertex 16                        \\number of vertices
property double x                        \\x coordinate of vertex
property double y                        \\y coordinate of vertex
property double z                        \\z coordinate of vertex
element face 32                          \\number of faces
property list uchar int vertex_indices   \\list of indices
property float gap_widths                \\manufacturability annotations
property int opposite_face_gap           \\manufacturability annotations
property int opposite_face_wall          \\manufacturability annotations
property float wall_thickness            \\manufacturability annotations
end header                               \\header end
\\vertex list: x,y,z coordinates
-13 -5 0
-13 -5 20
...
7 0 20
\\face list: vertex indices with manufacturability annotations
3 13 12 14 0.0 0 0 0.0                   \\no annotations
3 11 9 5 3.0 30 0 0.0                    \\gap annotation
...
3 11 7 15 0.0 0 0 0.0
```

Table 3.3: Object with manufacturability annotations in PLY format.

3.4.2 Mesh Visualizer

The mesh visualizer prototype integrated in the cloud platform is realized with the Web Graphics Library (WebGL) framework three.js [thr]. That visualizes the object with the manufacturability annotations in different color codes. Related to the consequences of the geometry guideline as defined in Section 3.3.2 has every consequence another color representation. This prototype represents not manufacturable faces in red and faces that need support structures in yellow. Annotations were added to get the exact information of the violated guideline. Ray Casting is used to select the 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 (see Figure 3.22). 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 (see Figure 3.22) to make the result more comprehensible.

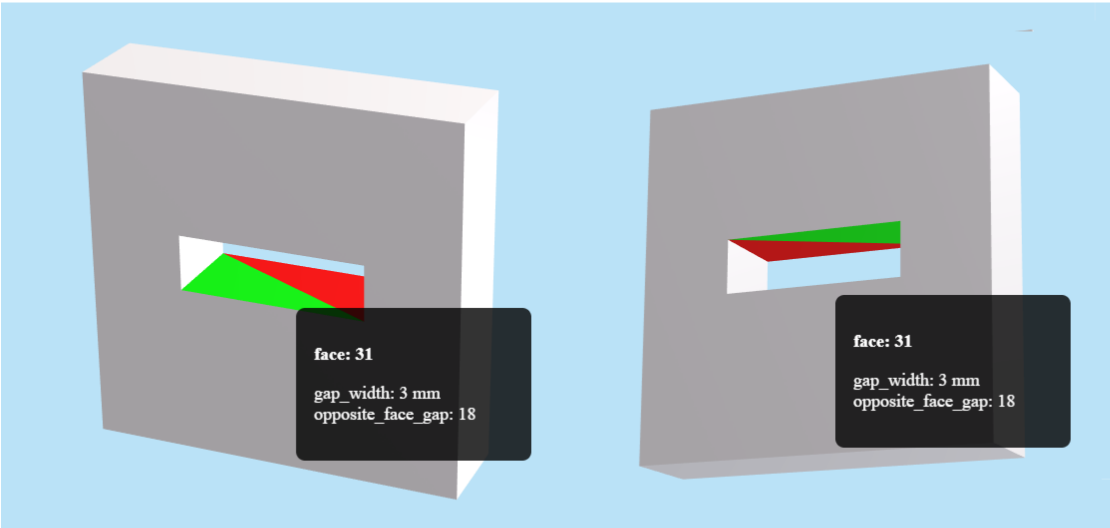


Figure 3.22: Mesh visualizer annotation

CHAPTER 4

Evaluation

This chapter covers the evaluation of the manufacturability system mesh processing with the mesh analyzer. The algorithms of Section 3.2 are evaluated for their applicability. Section 4.1 shows the evaluation with test workpieces that are designed for each guideline algorithm. Furthermore, the dependency of the algorithms on the resolution is discussed in this section. Another evaluation with the NIST test artifact used in the literature is described in Section 4.2. To further verify the applicability, tests were also carried out with industrial objects described in Section 4.3. The performance in terms of speed of the algorithms is another point on the applicability as AMMAS. Section 4.4 covers the runtime of the algorithms. However, the algorithms do not solve all problems and have their shortcomings listed in Section 4.5.

4.1 Evaluation with test manufacturable work pieces

This section describes the evaluation with objects that have been specially modeled to determine the printability of AM processes. A test object is created for each guideline. The test objects have the same structure as the objects of Rudolph and Emmelmann [RE17]. Other test objects for pin ratio, channel size along the z-axis, diameter and overhang in different types were also designed and tested.

4.1.1 Wall thickness

The wall thickness test object contains walls with the following thicknesses from left to right: 5, 4, 3, 2, 1, 0.8, 0.5, 0.3, 0.1 millimeters. The walls are positioned at a distance of five millimeters from each other (see Figure 4.1).

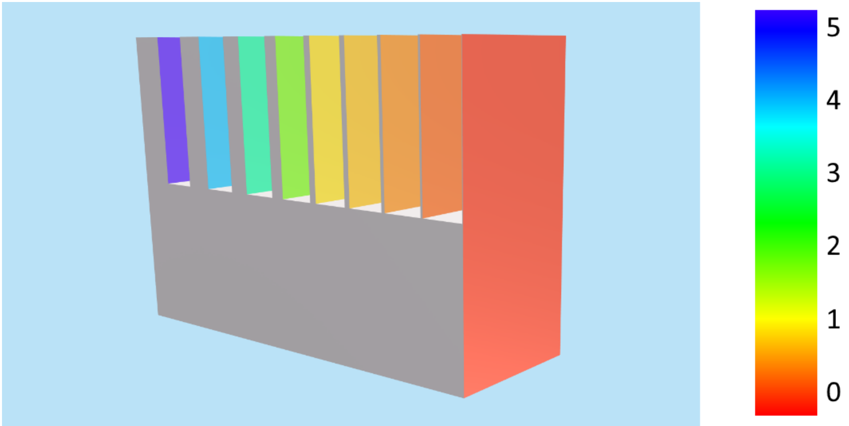


Figure 4.1: Testobject with different wall thickness in millimeters

4.1.2 Gap width

The gap width test object contains gaps with the following widths from left to right: 5, 4, 3, 2, 1, 0.8, 0.5, 0.3, 0.1 millimeters. The gaps are positioned at a distance of five millimeters from each other (see Figure 4.2).

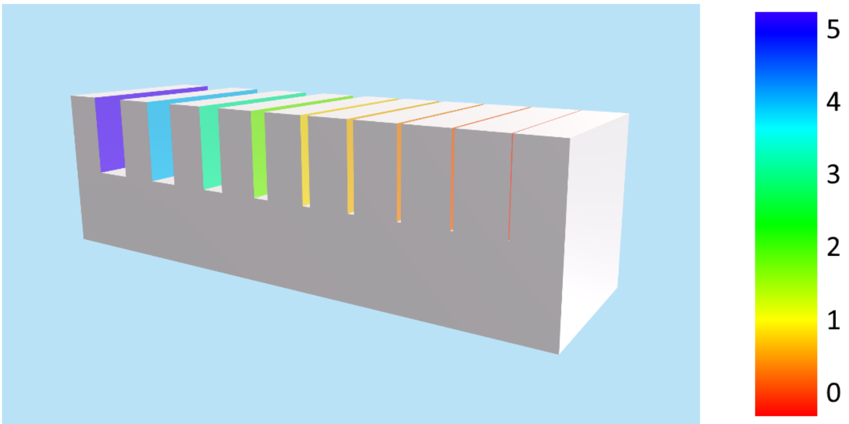


Figure 4.2: Testobject with different gap widths in millimeters

4.1.3 Pin radius

The pin test object has pins with the following radii from left to right: 5, 4, 3, 2, 1, 0.8, 0.5, 0.3, 0.1 millimeters (see Figure 4.3).

The ratio test is performed with the pin ratio object (see Figure 4.4 on the top left side) and the same pins with adapted heights from left to right: 20, 15, 12.5, 10, 8, 5, 2, 1, 0.5, 0.3, 0.1 millimeters (see Figure 4.4 on the bottom right side).

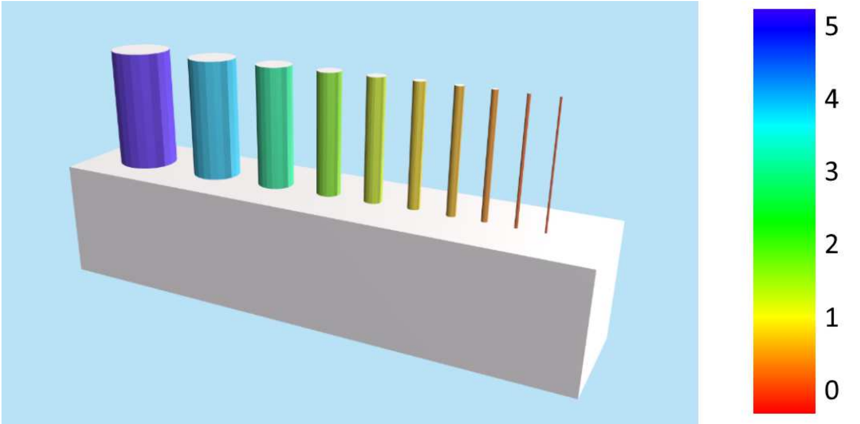


Figure 4.3: Testobject with different pin radii in millimeters

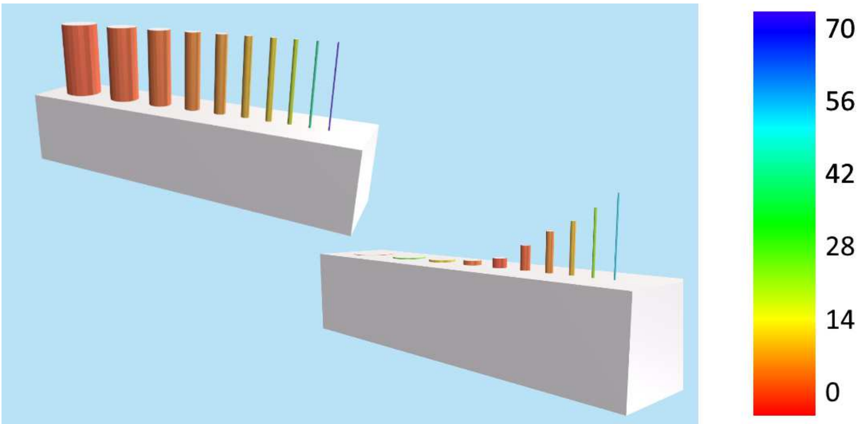


Figure 4.4: Testobject with different pin ratio

4.1.4 Hole radius

The test object contains holes with the following radii from left to right: 5, 4, 3, 2, 1, 0.8, 0.5, 0.3, 0.1 millimeters (see Figure 4.5).

4.1.5 Diameter, Channel size, Hole radius

Holes in AM have different manufacturability requirements depending on orientation and shape. A continuous hole is defined as a channel. It is possible to manufacture smaller channels of radii along the z-direction than the general hole radii indicated. Furthermore, holes require in the *Y-Z* or *X-Z* plane from a certain size on support structures. An automated AM manufacturability software must be able to recognize and differentiate these differences. The test object in Figure 4.6 shows the different possibilities. This object has channels and holes in different orientations and radii. Figure 4.6 illustrated the guidelines with a maximum hole diameter of four millimeters, a hole radius of three

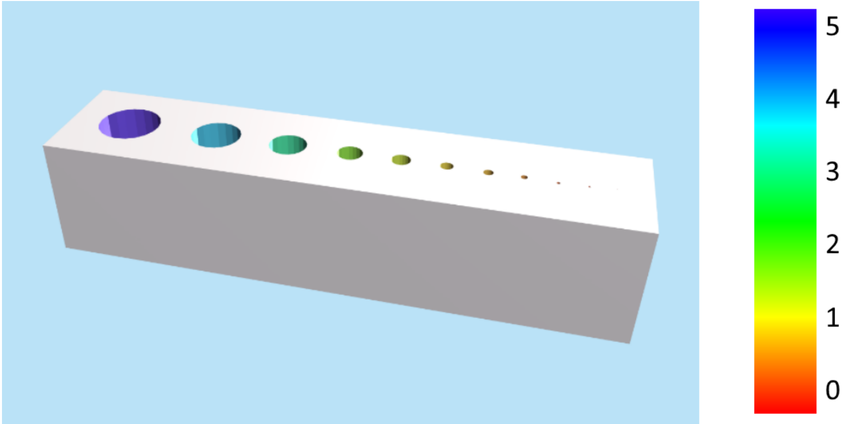


Figure 4.5: Testobject with different hole radii in millimeters

millimeters, and a channel radius of two millimeters.

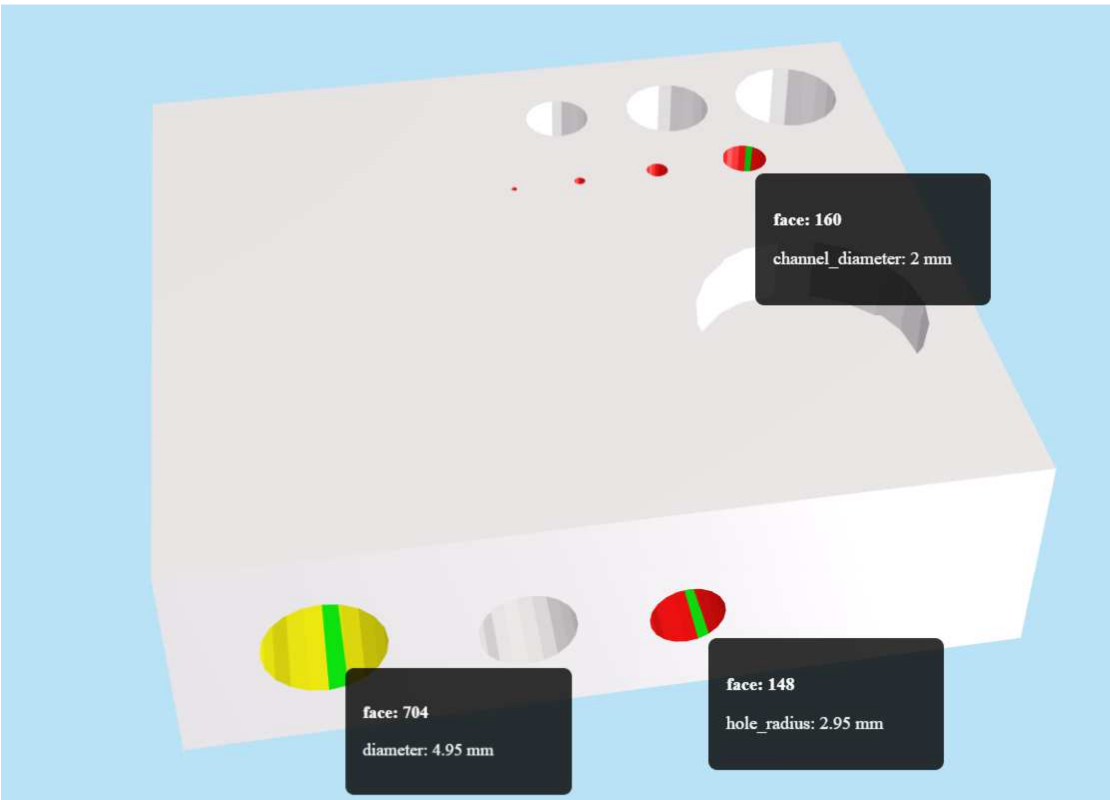


Figure 4.6: Annotated testobject with different holes (orientation, shape)

4.1.6 Overhang T-Type

The test object contains overhangs in T-type in different surface orientation from left to right: 5, 10, 15, 20, 25, 30, 35, 40, 45 degree (see Figure 4.7).

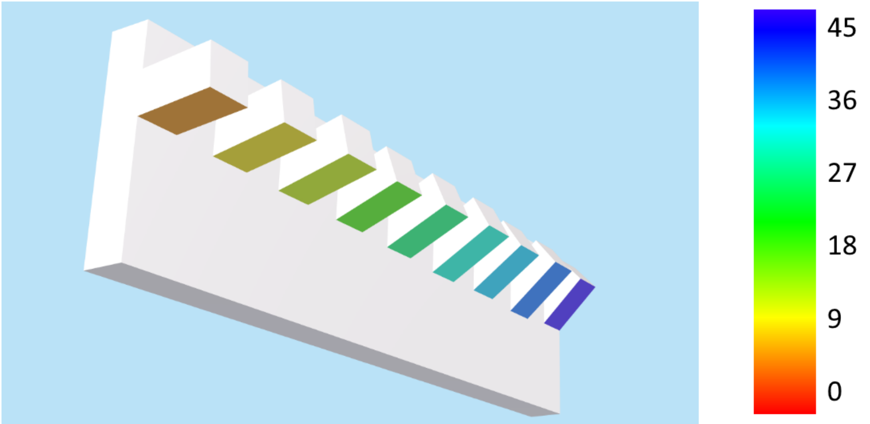


Figure 4.7: Testobject with T-type overhangs in different angles in degree

4.1.7 Overhang H-Type Length

The test object contains overhangs in H-type in different lengths from front to back: 20, 18, 16, 14, 12, 10, 8, 6, 4, 2 millimeters (see Figure 4.8).

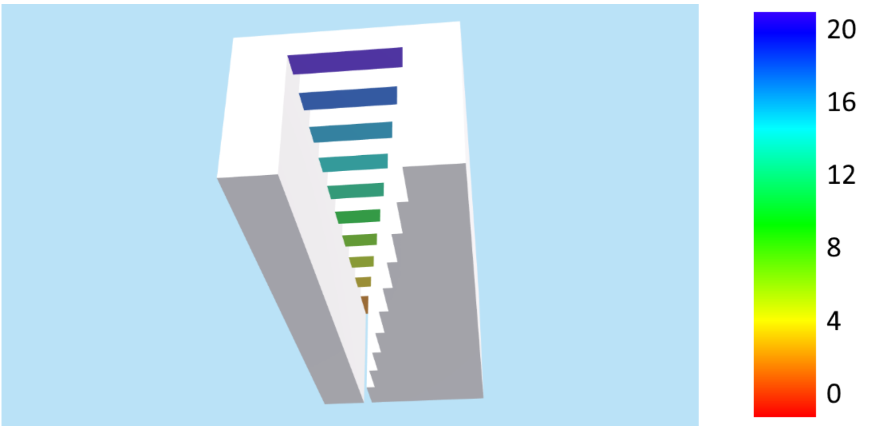


Figure 4.8: Testobject with H-type overhangs in different length in millimeters

4.1.8 Curvature in dependence of triangle resolution

The objects in STL format are constructed by triangles. With rectangular objects, the object is exactly transformed into a triangle mesh. However, curves with triangles could not be constructed exactly and are thus approximated. The approximation error becomes quadratically smaller with the object triangle resolution. An illustration of the error is shown in Figure 4.9. Therefore it is not possible to calculate the curvature of the

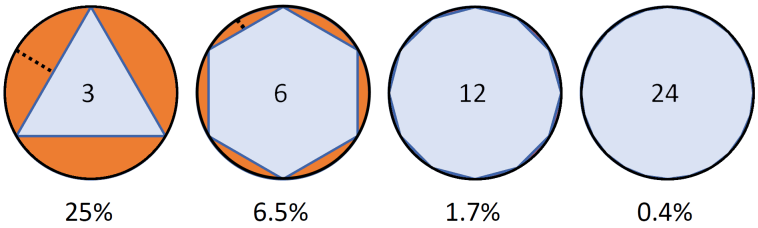


Figure 4.9: Approximation of smooth surfaces

object exactly. Furthermore, the formula used calculates the curvature between two adjacent triangles. A triangle has three adjacent triangles. So the calculation takes place in these directions and the most meaningful curvature for the manufacturability is the maximum curvature of the three. Taking the mean curvature or the Gaussian curvature as curvature value does not lead to any useful results for manufacturability. The values of the mean curvature and Gaussian curvature are calculated per vertex. The principal curvatures (minimum and maximum) can be calculated from the two. These values must then be interpolated for the triangular surfaces. CAD programs construct the triangle meshes irregularly. This means that as few triangles as possible are used to represent the object and the triangles have large size differences. Due to this kind of triangulation the values of the curvature are not usable. By calculating the curvature between adjacent triangles you can also determine the curvature of ellipses. The calculation of the radius of curvature of CAD-produced object roundings can have no uniform curvature between the triangles. This results in a radius range which represents the rounding of the object. Programs like Blender [ble] produce even curves with the *bevel* function. The curvature test was performed on different resolutions. The different resolutions are given in Figure 4.10 with the segment subdivision specification. The test result shows that the curvature calculation by the formula by Zhou et al. [ZYYW11] is independent of the resolution already from a segment subdivision of three segments.

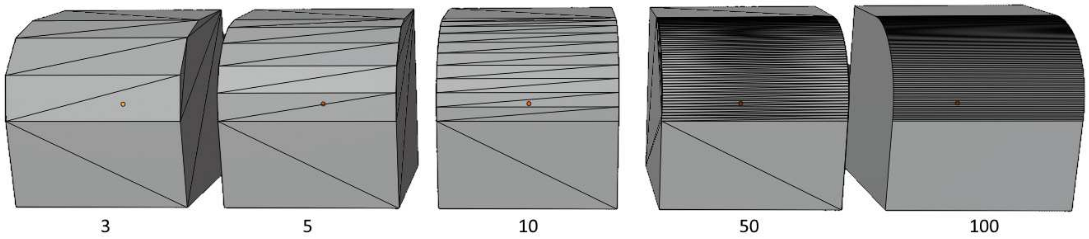


Figure 4.10: Rounded cube with different segment resolution from 3 to 100

4.2 Evaluation with NIST test artifact

glsnist has developed a test artifact to be proposed for standardization [MSC+14]. Figure 4.11 shows the schematic illustration of the test artifact. It has several features in a variety of sizes, locations, and orientations. This artifact is designed to test the limitations of an AMMAS. It can be used to evaluate the capabilities of AM processes and therefore

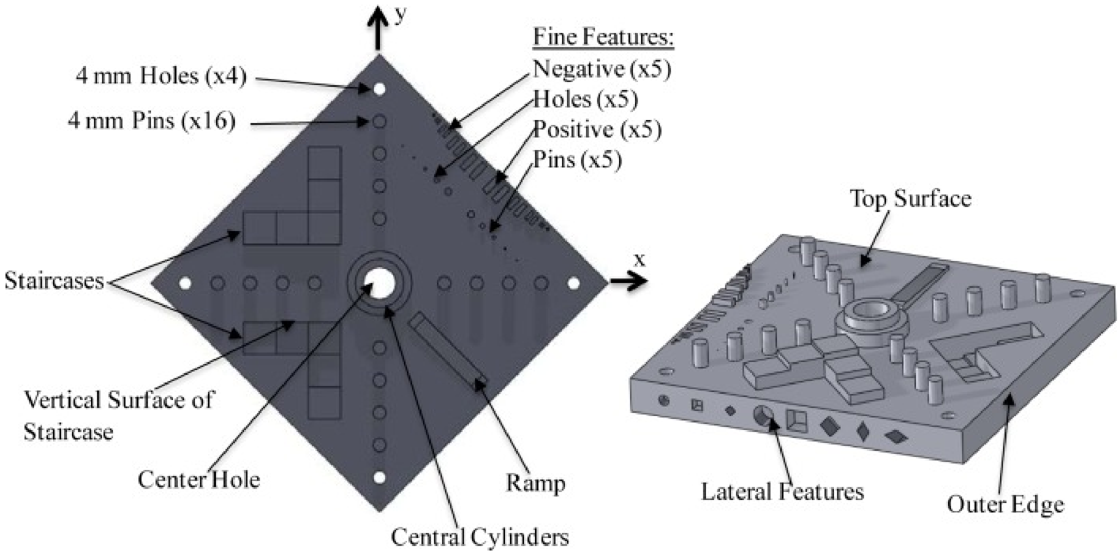


Figure 4.11: NIST test artifact showing a top view (left) and an oblique view (right) with annotations of important features [MSC+14].

it is suitable to test feature recognition algorithms and manufacturability check. The features of the artifact have sizes that go to the limit of manufacturability. Compared to other Manufacturability system as stated in Section 2.3, Shi et al. [SZB+18] evaluate their heat kernel signature with the NIST test artifact. Their approach separates the features of the test artifact and they assert that they correctly recognized all the features. This work does not categorize all features. Pins and holes are categorized and separated. The result of the mesh analyzer is shown in Figure 4.12. The color-coding on the top right side of Figure 4.12 shows radii in millimeters. All pin and holes are recognized and the radii are calculated correctly. The wall thickness and gap width algorithm are applied to the whole part and not on separated features. Due to the construction of the test artifact, a wall and gap threshold of two millimeters is chosen. So only features smaller than this threshold will be recognized. The ramps, as well as the complete object, are ten millimeters thick. In addition, the distances between objects are usually smaller than their dimensions. Thus, a wall thickness visualization with ten millimeters leads to visual clutter and is difficult to interpret. Figure 4.13 shows the result of the wall thickness and gap width algorithm with a threshold of two millimeters. The large area of the front edge is also recognized because the features were placed so close to the edge that the threshold was also undercut.

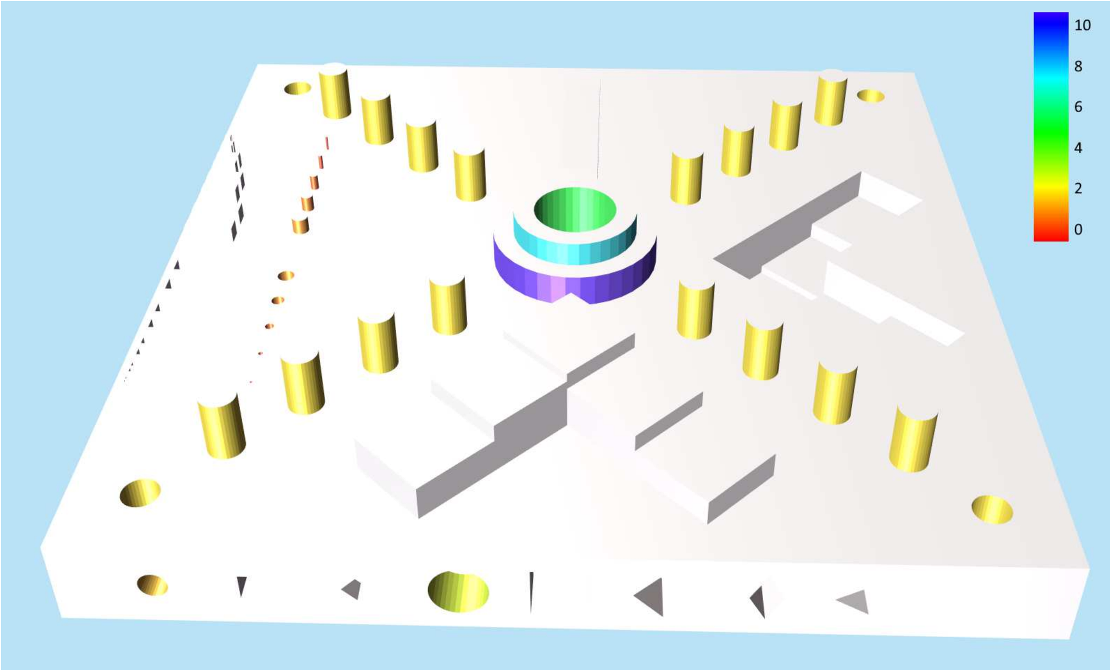


Figure 4.12: NIST test artifact with hole pin identification

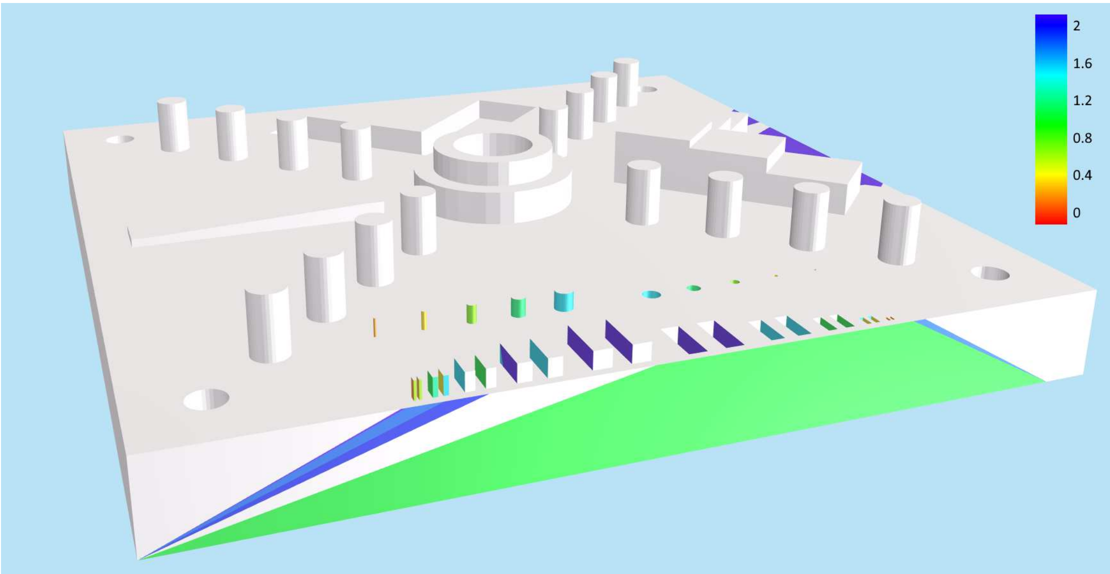


Figure 4.13: NIST test artifact with wall thickness and gap widths

4.3 Evaluation with industry objects

This section describes the evaluation of industrial objects. Three objects are evaluated exemplarily to show the possibilities of algorithms with more complex models. The evaluation is about the recognition of the forms of the structure and not about the immediate manufacturability. Guideline features such as holes, pins, and overhangs must be correctly identified. If the guideline features are correctly identified, the check of the manufacturability is only a comparison between the minimum feature value and the identified feature value. The first object represents a mounting fixture with two holes for the attachment for tapes (see Figure 4.14). This object contains two small fixation holes and an uneven hole for the tape. The underside of the bracket is also open and the rear panel has a straight edge for mounting. This object has several holes and actually consists of a large hollow pin. This work recognized six holes and the outside of the tape holder as a pin as it is shown in Figure 4.14.

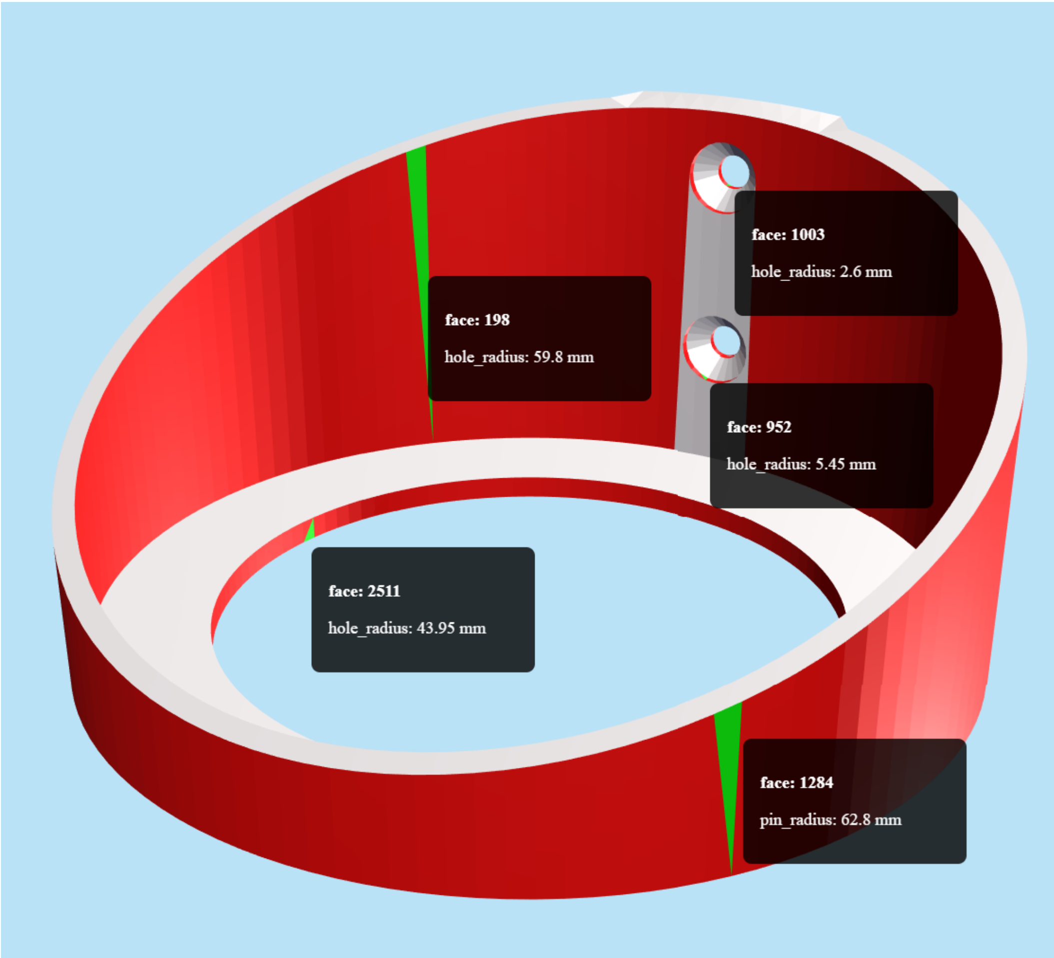


Figure 4.14: Tape holder with annotations

The second object is an antenna mount and frame spacer for the Twin Quad Frames CHopZaw (see Figure 4.15). This object 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 or four internal holes are easily identifiable and calculable features. However, the surrounding cylinder is not actually closed but already contains the connection of the overhang (see Figure 4.15). 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 clearly attributable to a T- or H-type due to its shape (see Figure 4.15). 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.

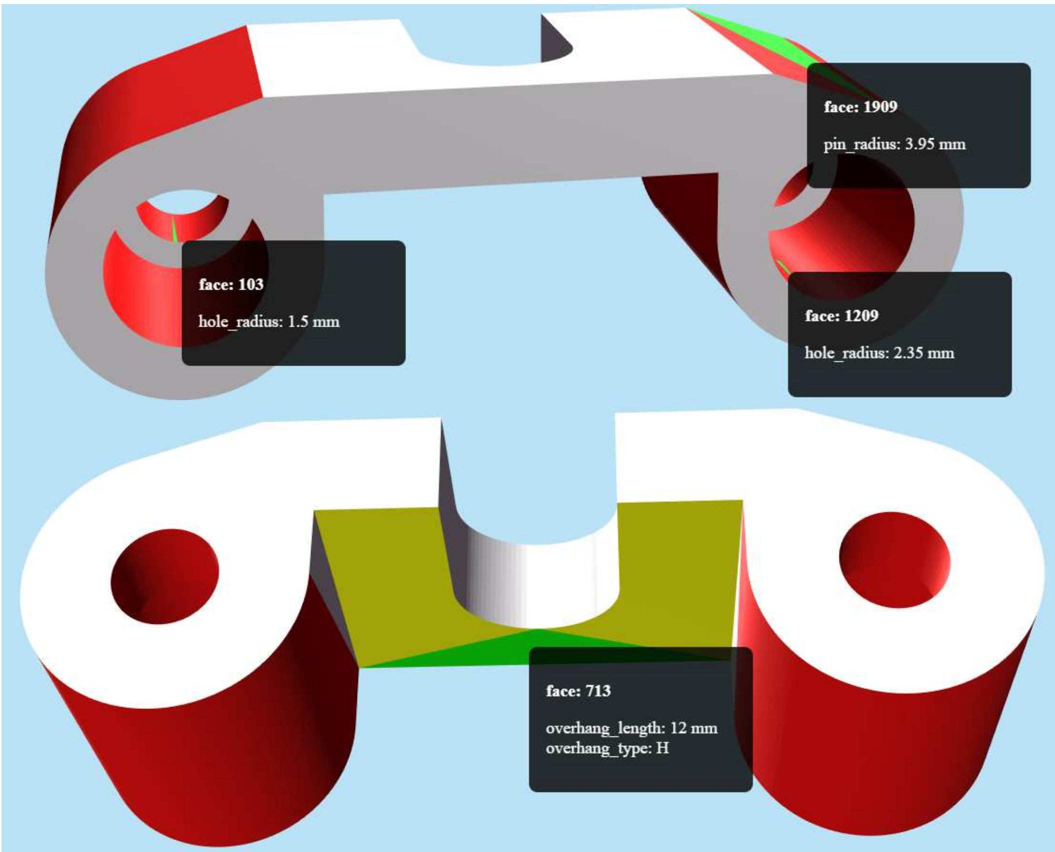


Figure 4.15: Antenna mount and frame spacer for the Twin Quad Frames CHopZaw with annotations

The third object represents a kind of a bench vice (see Figure 4.16). The vice contains all relevant features for the guideline check. It contains thin walls (see Figure 4.16 top left), small gaps (see Figure 4.16 top right), overhangs in H-type (see Figure 4.16 middle left), T-type (see Figure 4.16 middle right), a little rounding (see Figure 4.16 bottom left), a small channel and a large whole that need a support structure (see Figure 4.16 bottom right). All features are recognized and the dimensions were determined correctly.

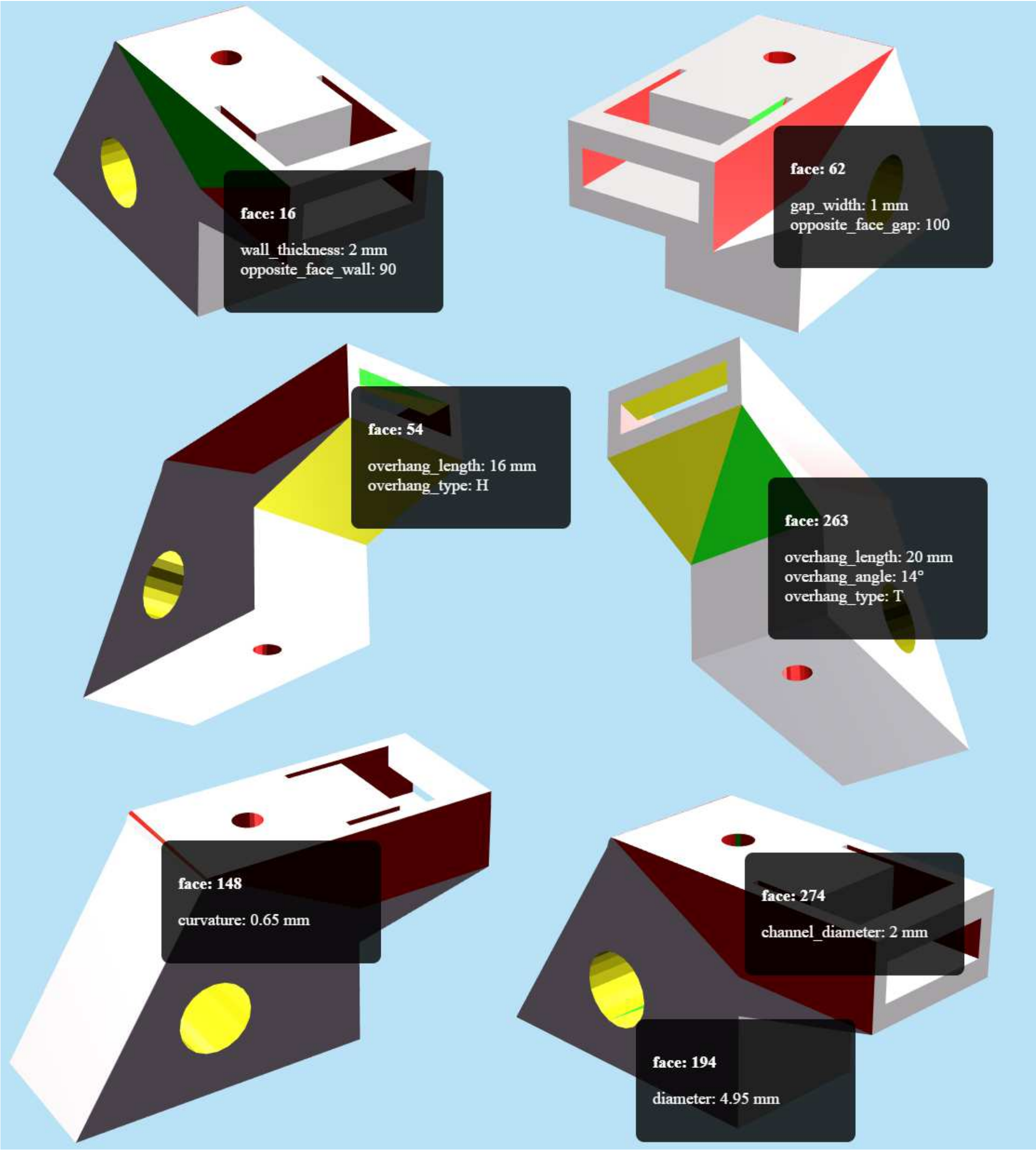


Figure 4.16: Kind of bench vice including all relevant guidelines

4.4 Algorithm runtime in dependence of the number of triangles

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 [Moh14]. In this work, the input is a triangle mesh and the runtime is analyzed in relation to the number of triangles. Table 4.1 gives an overview of the runtime of the algorithms.

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	$O(n)$	$O(n)$	$O(n)$

Table 4.1: Algorithm runtime in big-O notation

Wall thickness/gap width: The naive algorithm compares each triangle with each other. This results in an algorithm runtime of $O(n^2)$ in any cases. 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. Thus 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 no longer further investigated. In the best case, there are no circular structures in the object at all that correspond 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 3.7GHz. The runtimes of algorithms with the 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.

4.5 Shortcomings

The design of objects offers a high degree of freedom, which makes automatic checking using mesh processing algorithms more difficult. The presented approaches [TW16, RE17, SZB+18] mentioned that their further research incorporate more sophisticated object designs. Rudolph and Emmelmann [RE17] evaluates their approach only with test manufacturable workpieces. Shi et al. [SZB+18] uses the NIST test artefact and little adaptions for evaluation. Tedia and Williams [TW16] conducted their evaluation with a bunny and a teapot. This work was based on the evaluation of related literature but does not cover the treatment of more complex models. Nevertheless, some problems have already been identified based on industry objects (Section 4.3). The shape complexity makes it difficult to detect the necessary guideline features. Furthermore, even after successful extraction of the feature, 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 normal, but the calculation of the overhang requires more sophisticated algorithms. General, algorithms aimed at the recognition of feature structures are not yet applicable to all objects and must be further improved.
Besides, an efficient verification would be advantageous for the use of an AMMAS. Due to the processing based on triangles, the algorithm runtime is strongly dependent on the number of triangles. The runtime is explicitly discussed in Section 4.4. For the most time-consuming algorithm, a more efficient version has already been implemented using the k-d tree. However, the runtime is high and can be further reduced by parallel processing. Since many algorithms perform calculations with all triangles independently of each other, it is possible to transfer the algorithms to the GPU. The cloud- service-oriented model allows to use the resources of external data centers to reduce runtime due to better hardware.

CHAPTER 5

Conclusion and Future Work

AM makes it possible to create complex geometries in a wide variety of materials [NKK15]. Thus AM represents competition to the conventional manufacturing processes. Due to the novelty of the technology, the limitations of the technology have not been fully explored. Not every model that can be designed with a CAD can be manufactured using AM. This work deals with the rarely researched ceramic printing with the LCM process. To be able to print an object faultless, guidelines must be adhered to. Guidelines depend on the material, process and machine parameters used. This dependency is mapped in an ontology in a human and machine-readable format in such a way that simple queries provide the desired information. Together with the ontology, a cloud platform, and a mesh analyzer, the thesis presents an AMMAS. 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 AMMAS 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 like manufacturable or needs for a support structure. The evaluation with different objects in different complexity of the single guideline algorithms shows that the algorithms cannot solve the manufacturability yet completely automated. Simple structures can be efficiently checked for manufacturability using predefined algorithms. 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.

The constant further development in the area of AM increases the level of knowledge of manufacturability, which must also be further processed. 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.

List of Figures

List of Tables

List of Algorithms

Acronyms

AM Additive Manufacturing. xiii, 1, 2, 5–15, 21, 25–27, 29, 34, 38, 39, 44–47, 51, 53, 57, 65, 67, 69

AMMAS Additive Manufacturing Manufacturability Analysis System. 2, 26, 27, 29, 48, 51, 57, 63, 65

ASTM American Society for Testing Materials. 13, 40, 47

CAD Computer-Aided Design. 1, 15, 26, 27, 34, 56, 65

DF Design Fundamentals. 11

DFAM Design for Additive Manufacturing. 13, 14, 26, 67

DMDR Direct Manufacturing Design Rules. 9

DMDR 2.0 Direct Manufacturing Design Rules 2.0. 9

DP Design Principles. 11

DR Design Rules. 11

FDM Fused Deposition Modeling. 40, 44

GPR Guide-to-Principle-to-Rule. 11, 13

GPU Graphics Processing Unit. 19, 63

HKS Heat Kernel Signature. 21–23, 25, 67

ICAM Innovative Capabilities of Additive Manufacturing. 14

k-d tree k-dimensional tree. 31–33, 62, 63, 67

LCM Lithography-based Ceramic Manufacturing. xiii, 2, 12, 25, 26, 31, 33, 35, 38–40, 44, 65

NIST National Institute of Standards and Technology. xiii, xv, 3, 51, 57, 58, 63, 68

OWL Web Ontology Language. 39, 48

PLY Polygon File Format. 3, 49, 69

RDF Resource Description Framework. 42–44, 69

REST Representational State Transfer. 48

SDF Shape Diameter Functions. 16, 17, 67

SQWRL Semantic Query-enhanced Web Rule Language. 14, 25, 26

STL Standard Triangulation/Tesselation Language. 20, 21, 27, 30, 48, 55, 67

SVD Singular Value Decomposition. 22, 23, 25

WebGL Web Graphics Library. 49

Bibliography

[3dh] 3dhubs. `https://www.3dhubs.com/`. Accessed: 2019-08-20.

[AZ14] Guido AO Adam and Detmar Zimmer. Design for additive manufacturing—element transitions and aggregated structures. *CIRP Journal of Manufacturing Science and Technology*, 7(1):20–28, 2014.

[AZ15] Guido AO Adam and Detmar Zimmer. On design for additive manufacturing: evaluating geometrical limitations. *Rapid Prototyping Journal*, 21(6):662–670, 2015.

[ble] Blender. `https://www.blender.org/`. Accessed: 2019-10-20.

[BR17] Brian Garret Ben Redwood, Filemon Schöffer. *The 3D Printing Handbook*. 3D Hubs, 2017.

[BvHH+04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference. Technical report, W3C, http://www.w3.org/TR/owl-ref/, February 2004.

[DMR19] Direct manufacturing design rules. `https://dmrc.uni-paderborn.de/de/inhalt/forschung/interne-projekte/direct-manufacturing-design-rules-20/`, 2019. Accessed: 2019-09-20.

[DR17] Mahmoud Dinar and David W Rosen. A design for additive manufacturing ontology. *Journal of Computing and Information Science in Engineering*, 17(2):021013, 2017.

[EKG+15] Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Maxwell Perham, Jack Wileden, and Farhad Ameri. Knowledge management with an intelligent tool for additive manufacturing. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V01AT02A023–V01AT02A023. American Society of Mechanical Engineers, 2015.

[Eri05] Christer Ericson. Chapter 5 - basic primitive tests. In Christer Ericson, editor, *Real-Time Collision Detection*, The Morgan Kaufmann Series in Interactive 3D Technology, pages 125 – 233. Morgan Kaufmann, San Francisco, 2005.

[FGM+12] Ruth Felzmann, Simon Gruber, Gerald Mitteramskogler, Passakorn Tesavibul, Aldo R Boccaccini, Robert Liska, and Jürgen Stampfl. Lithography-based additive manufacturing of cellular ceramic structures. *Advanced Engineering Materials*, 14(12):1052–1058, 2012.

[GJRL18] S .Elaheh Ghiasian, Prakhar Jaiswal, Rahul Rai, and Kemper Lewis. From conventional to additive manufacturing: Determining component fabrication feasibility. In *Volume 2A: 44th Design Automation Conference*, page V02AT03A043, 08 2018.

[GRS+14] Ian Gibson, David W Rosen, Brent Stucker, et al. *Additive manufacturing technologies*, volume 17. Springer, 2014.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199 – 220, 1993.

[GZR+15] Wei Gao, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B Williams, Charlie CL Wang, Yung C Shin, Song Zhang, and Pablo D Zavattieri. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69:65–89, 2015.

[HKG18] Thomas Hagedorn, Sundar Krishnamurty, and Ian Grosse. A knowledge-based method for innovative design for additive manufacturing supported by modular ontologies. *Journal of Computing and Information Science in Engineering*, 18:021009–021009–12, 03 2018.

[HPR00] JungHyun Han, Mike Pratt, and William C Regli. Manufacturing feature recognition from solid models: a status report. *IEEE transactions on robotics and automation*, 16(6):782–796, 2000.

[Int15] International Organization for Standardization. *ISO/ASTM 52900:2015(en): Additive manufacturing — General principles — Terminology*. October 2015.

[JR19] Prakhar Jaiswal and Rahul Rai. A geometric reasoning approach for additive manufacturing print quality assessment and automated model correction. *Computer-Aided Design*, 109:1 – 11, 2019.

[JW17] Haeseong Jee and Paul Witherell. A method for modularity in design rules for additive manufacturing. *Rapid Prototyping Journal*, 23(6):1107–1118, 2017.

[KHE15] Jannis Kranz, Dirk Herzog, and Claus Emmelmann. Design guidelines for laser additive manufacturing of lightweight structures in tial6v4. *Journal of Laser Applications*, 27(S1):S14001, 2015.

[KRWK18] Samyeon Kim, David W Rosen, Paul Witherell, and Hyunwoong Ko. Linking part design to process planning by design for additive manufacturing ontology. In *Proceedings of the 3rd International Conference on Progress in Additive Manufacturing (Pro-AM 2018),*, pages 303–308, 2018.

[KRWK19] Samyeon Kim, David W Rosen, Paul Witherell, and Hyunwoong Ko. A design for additive manufacturing ontology to support manufacturability analysis. *Journal of Computing and Information Science in Engineering*, 19(4):041014, 2019.

[LDAZ16] Tobias Lieneke, Vera Denzer, Guido A.O. Adam, and Detmar Zimmer. Dimensional tolerances for additive manufacturing: Experimental investigation for fused deposition modeling. *Procedia CIRP*, 43:286 – 291, 2016. 14th CIRP CAT 2016 - CIRP Conference on Computer Aided Tolerancing.

[LTM+18] 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. 2018.

[Lu16] Tianxiang Lu. Towards a fully automated 3d printability checker. In *2016 IEEE International Conference on Industrial Technology (ICIT)*, pages 922–927. IEEE, 2016.

[MBC12] Jaehwan Ma, Sang Won Bae, and Sunghee Choi. 3d medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer*, 28(1):7–19, 2012.

[Moh14] Austin Mohr. Quantum computing in complexity theory and theory of computation. *Carbondale, IL*, 2014.

[mor19] Image processing: Morphological operations. `https://medium.com/@aishwariyag96/image-processing-bef9cec13e4d`, 2019. Accessed: 2019-09-20.

[MSC+14] Shawn Moylan, John Slotwinski, April Cooke, Kevin Jurrens, and M Alkan Donmez. An additive manufacturing test artifact. *Journal of research of the National Institute of Standards and Technology*, 119:429, 2014.

[MW15] Nicholas Alexander Meisel and Christopher Williams. An investigation of key design for additive manufacturing constraints in multimaterial three-dimensional printing. *Journal of Mechanical Design - Transactions of the ASME*, 137(11), 11 2015.

[MWJ17] Mahesh Mani, Paul Witherell, and Haeseong Jee. Design rules for additive manufacturing: A categorization. 2017.

[NKK15] Saigopal Nelaturi, Walter Kim, and Tolga Kurtoglu. Manufacturability feedback and model correction for additive manufacturing. *Journal of Manufacturing Science and Engineering*, 137(2):021015, 2015.

[num] Numpy. `https://numpy.org/`. Accessed: 2019-08-20.

[Pic12] Mike Pickton. What is a voxel, anyway? voxels vs. vertexes in games. `https://www.fujixerox.com/eng/company/news/release/2016/001281`, 2012. Accessed: 2019-09-20.

[pro] Protégé ontology editor. `https://protege.stanford.edu/`. Accessed: 2019-04-10.

[pyt] Python. `https://www.python.org/`. Accessed: 2019-08-20.

[RE17] Jan-Peer Rudolph and Claus Emmelmann. Analysis of design guidelines for automated order acceptance in additive manufacturing. *Procedia CIRP*, 60:187–192, 2017.

[ros] Rosetta ontology. `https://github.com/jacekmalec/Rosetta_ontology/`. Accessed: 2019-04-10.

[SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[SSMM18] Uwe Scheithauer, Eric Schwarzer, Tassilo Moritz, and Alexander Michaelis. Additive manufacturing of ceramic heat exchanger: opportunities and limits of the lithography-based ceramic manufacturing (lcm). *Journal of Materials Engineering and Performance*, 27(1):14–20, 2018.

[SZB+18] Yang Shi, Yicha Zhang, Steven Baek, Wout De Backer, and Ramy Harik. Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications*, 15(6):941–952, 2018.

[thi] Thingiverse. `http://thingiverse.com/`. Accessed: 2019-08-20.

[thr] three.js. `https://threejs.org/`. Accessed: 2019-08-20.

[TLWZ18] J. Tominski, Stefan Lammers, Colin Wulf, and Detmar Zimmer. Method for a software-based design check of additively manufactured components. In *Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium*, 2018.

[TW16] Saish Tedia and Christopher B Williams. Manufacturability analysis tool for additive manufacturing using voxel-based geometric modeling. In *27th Annual International Solid Freeform Fabrication (SFF) Symposium*, 2016.

[TZ16] Yunlong Tang and Yaoyao Fiona Zhao. A survey of the design methods for additive manufacturing to improve functional performance. *Rapid Prototyping Journal*, 22(3):569–590, 2016.

[Ull10] David G Ullman. The mechanical design process: Part 1, 2010.

[WRWS15] Dazhong Wu, David W Rosen, Lihui Wang, and Dirk Schaefer. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer-Aided Design*, 59:1–14, 2015.

[Zho] Qingnan Zhou. Pymesh. `https://pymesh.readthedocs.io/en/latest/`. Accessed: 2019-08-20.

[ZYYW11] Shilin Zhou, Jianping Yin, Xiaolin Yang, and Junyun Wu. Accurate curvature approximation of 3-dimension discrete points. In *Communication Systems and Information Technology*, pages 803–810. Springer, 2011.