[35] SteveFortune (2008):"Fortune's 2D Voronoi diagram code", http://www3.cs.stonybrook.edu/~algorith/implement/fortune/implement.shtml.

[36] Weisstein, E. W.: "Voronoi Diagram", *From MathWorld–A Wolfram Web Resource*, http://mathworld.wolfram.com/VoronoiDiagram.html.

Parte II

Anexos

Anexo A

Versión original del proyecto en inglés

Technische Universität München
Fakultät für Maschinenwesen
Lehrstuhl für Mikrotechnik und Medizingerätetechnik
Univ.-Prof. Dr. Tim C. Lüth

Master Thesis

MATLAB-integrated Software Implementation of Tool-Path Strategies for droplet-based 3D Printers

Javier Sobreviela

Matr.-Nr.: 03650554

Betreuender Hochschullehrer:	Univ.-Prof. Dr. Tim C. Lüth
Betreuerin:	Jelena Prša, M.Sc. (Hons)
Ausgegeben am:	02.05.2014
Abgegeben am:	31.10.2014

Ehrenwörtliche Erklärung

(Deutsch)
Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht.

(English)
I solemnly declare that I have prepared this thesis independently and without use of any not specified aids; directly or indirectly acquired thoughts from external sources (including electronic sources) are without exception identified as such.

Garching bei München, den 31.10.2014

Javier Sobreviela

Acknowledgements

I'm thankful the university of Zaragoza and the Technische Universität München for the opportunity of fully developing my studies in industrial engineering. I want to thank also my teachers and directors that have helped me in this process of academic formation.

I want to thank my family and friends for their support along these years. Without them it would have been much more difficult.

Finally I want to express my gratitude to my adviser and the MiMed chair of the TUM for the opportunity of developing this thesis.

Garching bei München, Oktober 2014 Javier Sobreviela

Contents

Part I

Introduction

Chapter 1

Application

Rapid prototyping (RP) is defined as a collection of technologies that are driven by computer aided design (CAD) data to produce physical models and parts through an additive process [1]. Today, a variety of different additive manufacturing (AM) processes are available. They are measured and characterized through different parameters, as: geometrical tolerance, applied material, time elapsed, working volume, movement freedom and many others.

A vast majority of the AM processes are also described as layer manufacturing (LM) processes, because they produce the object layer by layer, from bottom to top or vice versa. In these cases the material is added to the actual layer by a machine. The machine lays the material following the data contained in a file that is taken as input. This file is the result of the software treatment of the CAD model of the object. This technology is currently used in a variety of fields, for example:

- Mold building [2]: production time and cost reduction in the fabrication of molds that otherwise should be machined.
- Aerospace [3, 4, 5]: lighter, more customizable and cheaper parts.
- Medical devices and Prosthetics [6, 7, 8]: more customizable and cheaper prosthetics, operations planning.
- Automotive industry [9]: rapid prototyping models used in the development of final pieces.

In the papers referenced the advantages of AM are the production time and cost reduction and the capacity of producing more complex forms that in other fabrication processes would be more difficult. Although AM is not perfect and has also its own limitations, as will be elaborated further on in this report. The overcoming of this limitations may lead to additional applications.

Chapter 2

State of the Art

Additive manufacturing is a wide field with many areas to focus on for improvement. In this chapter an overview on the technologies available will be presented, comprising the common phases that these processes follow to produce physical models. After this introduction, the research conducted over the state of the art will be exposed.

2.1 Additive Manufacturing Overview

Before going into more specific topics is deemed advisable to go through an overview of the technologies and processes that govern this field of engineering in order to acquire the knowledge that allows the user to understand the implications of each part into the whole.

In this summary two areas will be covered. The existing fabrication technologies available nowadays for the use in industry. And the general process common to all of these applications that goes from the design to the physical model.

2.1.1 Additive Manufacturing Technologies

As stated before AM includes a diversity of technologies. These are classified through the analysis of diverse characteristics, including process type, material type, size, cost, or ease of operation [1]. In the following lines two different classifications will be explained to give an overview of the today's available systems.

Classification by process type

In [1] a classification is proposed according to the process type and four technologies are discussed. It should be taken into consideration that these are not the only processes available, but the most extended in industry. As a rather new field

of engineering the progress is quick and in the last times new technologies keep appearing, offering new possibilities.

Stereolithography Stereolithography (SLA) is a photopolymerization [10] process that makes use of radiation curable resins, which when irradiated with ultraviolet wavelength or some visible light undergo a chemical reaction to solidify. The common configuration of these systems, as represented in figure 2.1, involve a vat containing the photopolymer and a certain configuration of radiation generating components, as may be lamps or lasers, the necessary optics and a directing system, that can be a scanning mirror group or the possibility of moving the radiation elements.

The fabrication process of a model is as follows. Inside the vat a platform that moves in the Z-coordinate sustains the object that is being generated layer by layer as the radiation beam is directed according to the desired geometry. Once one layer is finished and solidified over the platform, it moves down to allow the next layer of photopolymer to be solidified.

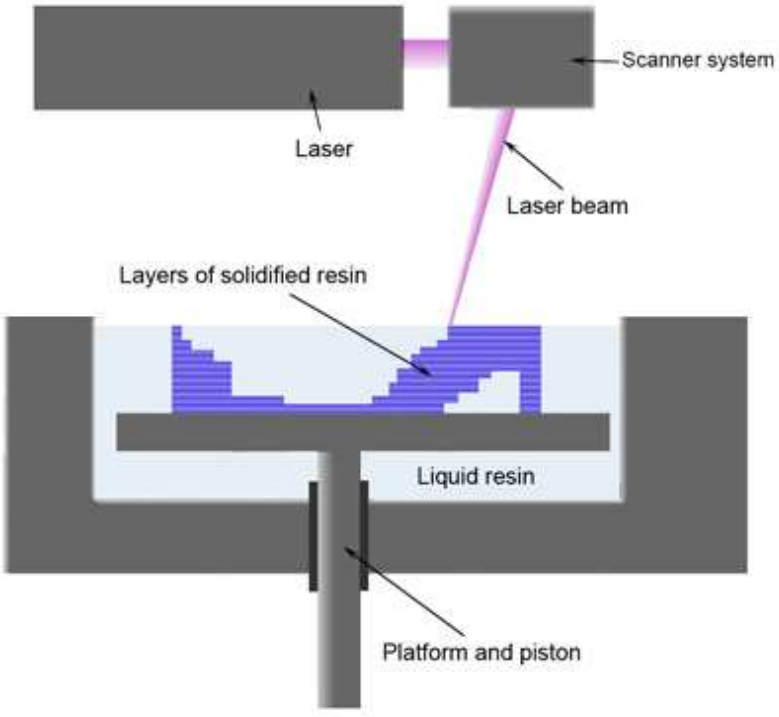


Figure 2.1: Stereolithography Apparatus, [11]

These systems have prices between $180000 and $800000 [1, 2004] and offer a good balance of speed and quality making it a good technology for a broad range of applications.

Selective Laser Sintering Selective laser sintering (SLS) is a powder bed fusion process [10] using not only plastic materials but also metals and ceramics. As the name implies the material is initially in a solid powder form. Using one or more thermal sources fusion is induced between the powder particles. The system, figure 2.2, is composed of heat source, which is commonly a laser with scanning mirrors, a build platform, which carries the model and a feeding subsystem, composed of the feeding cartridges that contain the powder, which will be deployed over the platform on each new layer by a roller. To reduce the laser energy consumption an infrared heater can be used to elevate the general temperature of the material.

Similarly to SLA, in this case the laser sinters each layer of green material at a time. For a new layer the platform moves down vertically leaving space that will be filled with powder in a roller pass. Being in solid state the not-sintered powder serves as support for complex geometries.

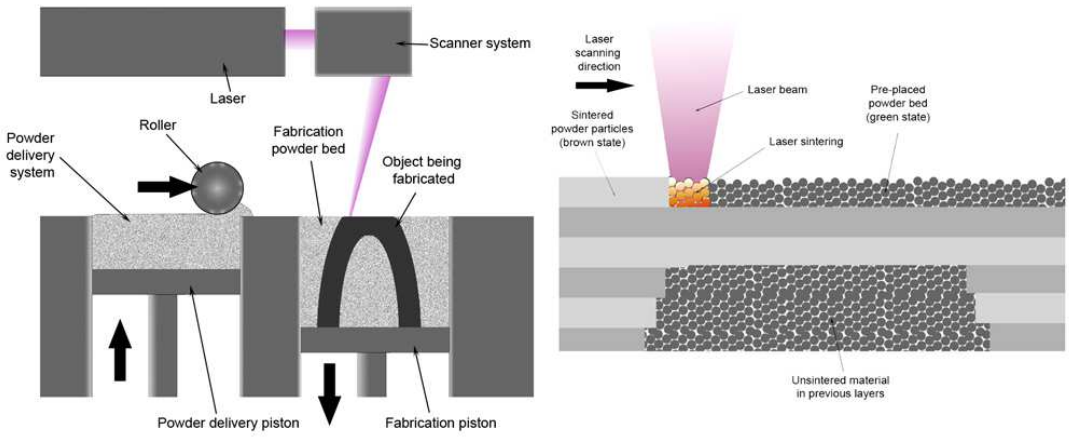


Figure 2.2: Selective Laser Melting System, [12]

An average price for these type of systems would be $300000 [1, 2004]. The biggest advantage of this process is the variety of materials that can be utilized.

Fused Deposition Modeling Fused deposition modeling is an extrusion-based process [10] that uses semi-solid state material contained in a reservoir forcing it through a nozzle when applying pressure. With constant pressure the material

will flow forming roads of constant cross-sectional diameter. It will now solidify maintaining the shape and bonding to the already extruded material, producing as result a solid structure.

The parts from this kind of systems are usually divided in two sections, the reservoir and deposition mechanism. The reservoir is the source of fused material to be deposited. Generally it is a supply of solid material that is fused in the amounts required, like a roll of plastic wire or a classical injection machine working with plastic granulate. The deposition mechanism comprises the pressure application system, the constant diameter nozzle and the bed or platform. To allow the building of 3D objects an x-y-z relative movement should exist between bed and nozzle. An example of the system concept is depicted in figure 2.3.

The operation is usually as follows. The solid supply is carried through a melting chamber leaving it via the nozzle. It is then laid on the bed as it or the nozzle moves producing the filament of material. As the nozzle completes one layer, it will move vertically before beginning with the next one.

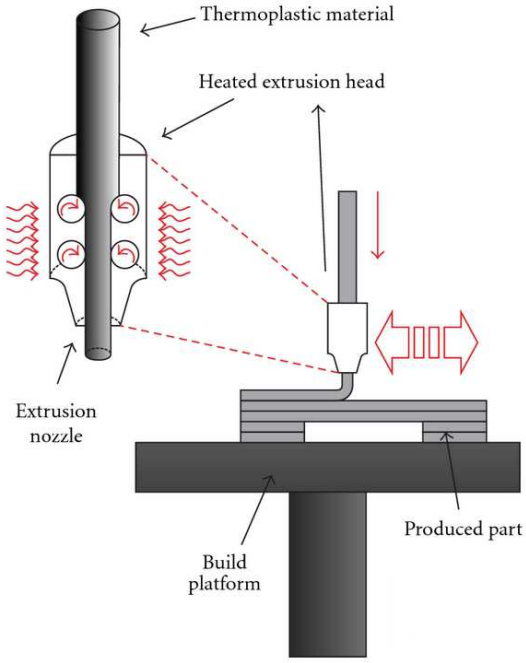


Figure 2.3: Fused Deposition System, [13]

Large systems range in $150000-250000 [1, 2004]. But the fairly mechanical simplicity has made possible the development of more accessible machines, which even without offering the same final quality as the large ones, can be obtained and used by a more broad community, like students or small offices. An example

of this smaller machines is the MakerBot [14]. The "Replicator Mini" model is currently available for a price of $1375 [14, 2014]

Classification by material

A more general classification that covers almost all the numerous RP systems available in the market is proposed in [15] based on the initial form of the material employed. The three categories are: liquid-based, solid-based and powder-based.

The liquid-based RP systems function curing an initially liquid material, converting it to the solid state. An example of curing process would be the previously explained photopolymerization, used in SLA.

The solid-based RP systems can be separated in two subgroups. The ones that work with the melting and solidifying method, like FDM, and those based on the cutting and gluing method, i.e. LOM (Laminated Object Manufacturing).

The powder-based systems, although being solid material, are defined as their own group because of its extension, defining those that use the material in an initial grain-like form. All of these systems employ the method known as joining/binding. SLS is a member of this group.

2.1.2 Additive Manufacturing General Process

The existing AM technologies share a general process for converting a 3D computer model into a physical object. Two stages are easily differenced in this process: the software stage and the fabrication stage. In the next lines both will be explained.

Software stage

This stage involves all the steps that produce a comprehensible and correct data file, which allows the machine to fabricate the desired physical model. In the literature a variety of sequences for this stage are proposed [1, 10, 16]. They can be summarized in:

1. CAD model design
2. CAD to STL conversion
3. STL file verification and fixing
4. STL slicing
5. Build data generation

The input of this process is the CAD file that contains the design of the object. At this point of the design two misconceptions arise amongst new users, as presented in [17]. The first one is that RP requires a closed volume of the object and the next one is that what is being designed in the CAD software is not what will finally result. This is the effect of the limitations of these processes. For an optimal use of these technologies, those limitations should be taken into account at the design step.

The next step is to translate the tessellated data of the CAD object into a triangle mesh of the object shell contained in a STL file. This step implies an estimation of the model, in view of the fact that the bigger the triangles the worse the approximation to the original geometry. This STL format is the de facto standard in the industry [15] and it allows further manipulations in subsequent steps.

Once the STL file is produced doing verification is in order to check if fixing is needed. In industry a significant amount of files are found to be corrupt [1]. There are automated software implementations that analyze the data and, if necessary, perform repairing operations to allow the file to be comprehended by the succeeding phases.

Now orientation and placing must be decided. Then the model undergoes a slicing process that divides it in a vertical manner. The height of these slices is set by the vertical or Z-coordinate resolution. A reduction in the layer thickness yields a higher geometrical accuracy, but on the other side the building time increases considerably. This is the "stair effect", which will be presented later on. A balance between surface quality and build time should be made.

The result of the slicing are the 2D contours that form the object along its height, as they are the intersections of the object with the slicing planes. There can be inner and outer contours since there may be empty spaces inside the part. The surface contained between inner and outer contours of each slice is then the infill area. As the last step of the stage an infill strategy is to be applied to fill this area. This strategy is itself a process that takes different inputs, as the speed of the nozzle movement, the path width to be laid or the allowed overlap between paths laid. This strategy will produce then the toolpath that the machine will follow to fill each layer, producing the real object.

Fabrication stage

After the data is prepared and transmitted to the machine there are only two left phases to accomplish. First of all the actual fabrication is done by the machine and is actually almost fully automated as the machine follows the introduced data, being the job of the operator to feed the material and to extract the fabricated model.

In some cases this fabricated model is not finished and needs further treatment. A finishing phase is required to finally produce the desired physical model. This comprises a series of operations:

- Removing the support structures used.
- Piece cleaning: in some technologies non-processed material has to be removed.
- Mechanical treatments: as polishing or sanding to improve the surface quality.

At the end the model or prototype is finished and ready for use

2.2 System description

In the previous section the most used technology was presented. In this thesis the system considered is a bit different from the common machines previously shown. In this section the specifics of this system will be presented, regarding mechanical function and software implementation. All the developments of this thesis will be primarily applicable to this system.

2.2.1 Mechanical functioning and explanation

In this thesis the system employed is based in fused deposition modelling (FDM). The most commonly used machines for FDM are based on filament printing [18]. While also using fused plastic material as printing material this case process is based on drop printing. In the following lines the machine's functioning process and the different parts will be described and explained.

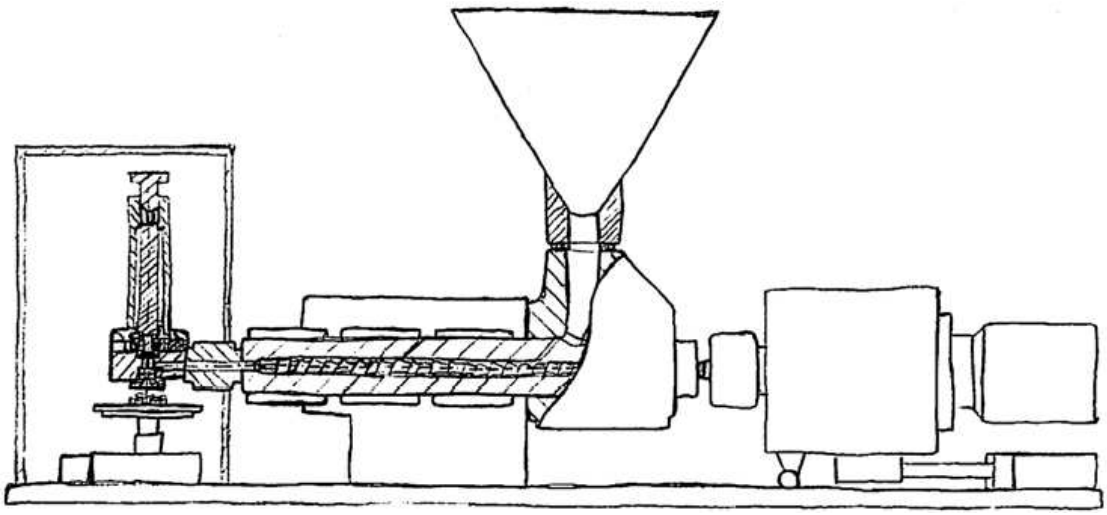


Figure 2.4: Sketch of the mechanical arrangement [Hehl,2010]

The machine, represented in figure 2.4, is a drop-based plastic deposition 3D printer and is divided in two main parts: the processing unit and the deposition unit.

The processing unit is nothing else than the injection part of a common plastic-injection machine. The goal is to fuse the plastic granulate preparing it for the deposition. The material introduced through the hopper is carried by the rotating screw through the heated barrel enabling the plastic to fuse and be injected at the end of the barrel to the deposition unit, presented in figure 2.5.

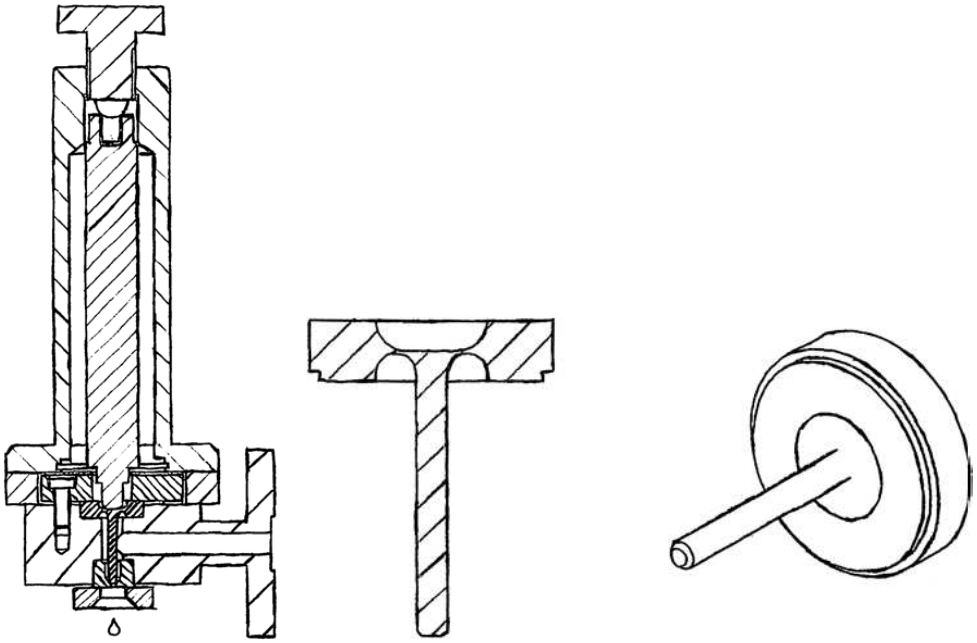


Figure 2.5: Deposition unit and flexible needle [Hehl,2010]

Once the fused plastic enters the deposition chamber the needle that closes the nozzle is displaced through a piezoelement actuating at the flexible part of the needle piece. As it moves upwards the plastic can flow down producing a drop. The duration of this opening is measured in milliseconds. The drop temperature and size should be enough to ensure the adhesion forces.

Another difference with the common FDM machines is that in our case the deposition unit is fixed in a certain position. Therefore to produce the relative movement is the platform the one that moves to produce the geometry of the piece to be printed. Displacements are available in all 3 dimensions. The concept is displayed in figure 2.6.

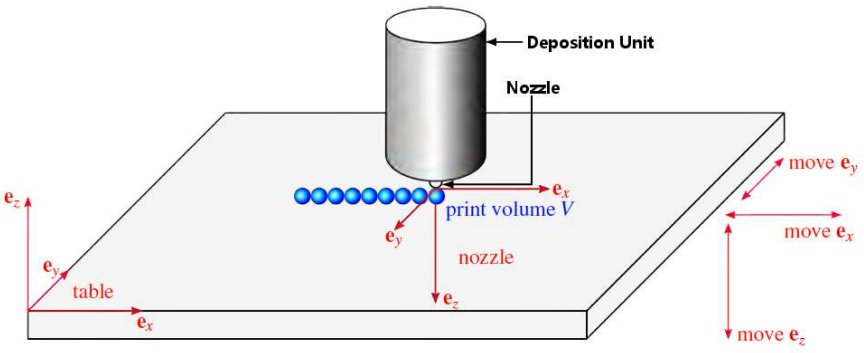


Figure 2.6: Machine platform and relative movement, [19]

2.2.2 Software Environment and G-Code Generation

The developments and products of this thesis work over a preexisting software environment that processes STL files and generates G-Code that the machine can understand in order to manufacture the pieces.

This software is specific for the application on the machine this thesis is working with. The general workflow goes as following:

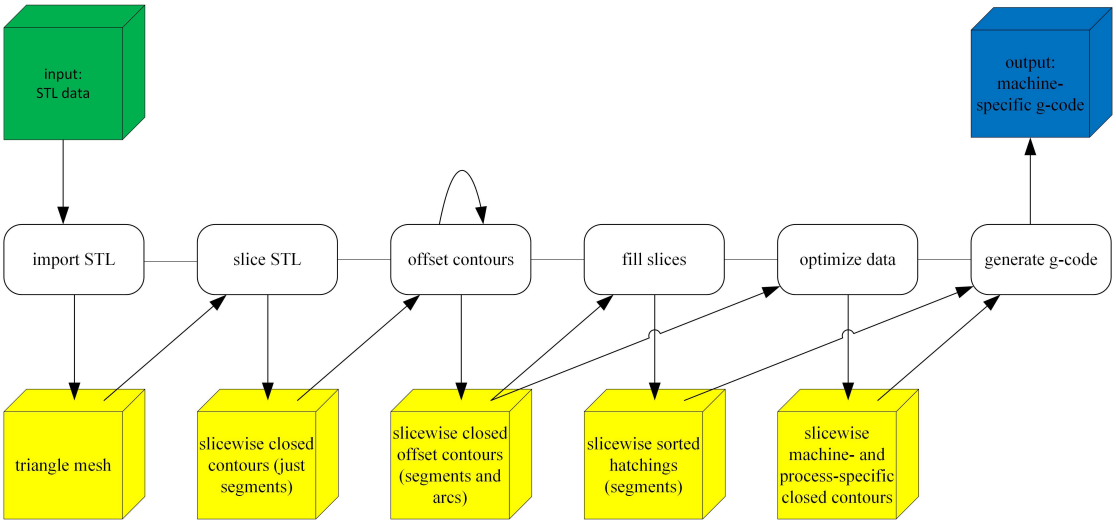


Figure 2.7: Workflow showing input data (green), temporary data (yellow) and output data (blue), [19]

In short, the STL data is read and the triangle mesh is sliced according to a certain Z-Coordinate step. This slicing produces a set of contours that are going to be processed. The next step is to compute the tool path for the filling of the

slice and following an optimization of this data. Both last steps will conclude in a generation of the G-Code that can be now passed to the machine to print. A more extensive explanation of the workflow can be found in [20, 19].

The G-Code is composed of the orders that the machine understands and process in order to build the object. This orders comprise a command and the 3D coordinates where that command is going to be executed. The command states which type of printing is to be executed.

2.3 Filling methods research

As explained previously, after slicing the model as represented in the STL data file, the next step is to generate the filling of those slices. This data represents the information the machine needs to actually build the object. Being the focus of this thesis, this phase of the software treatment will be covered extensively on this section through the research of the state of the art in this specific field.

2.3.1 Definition

In the literature the slice filling procedures are known as tool-path strategies [21, 22], path planning [23] or layer filling strategy [24]. This path is the trajectory that the AM tool, be that a depositing nozzle or a curing laser, follows across the layer to fabricate the object. The goal of this processing stage is to develop this path in relation to the geometrical data present in the slice and the desired characteristics of the object. These characteristics include surface quality, inner filling, as it can be solid or a sparse structure for example, building time and stiffness.

2.3.2 Strategies

Currently two major strategies gather the most part of the research: hatching or raster filling and contour filling [25, 23]. There are also other proposed strategies but those present more limitations and their use is really constrained.

Hatching filling

Raster or hatching filling, represented in figure 2.8, also called zigzag filling [21, 22], consist in a back and forth motion following a raster defined in the inside of the slice perimeters. This raster is composed of two kind of roads [26], the vector segment, that follows the direction of the raster and the turn segment that joins the end of a vector segment with the beginning of the next segment.

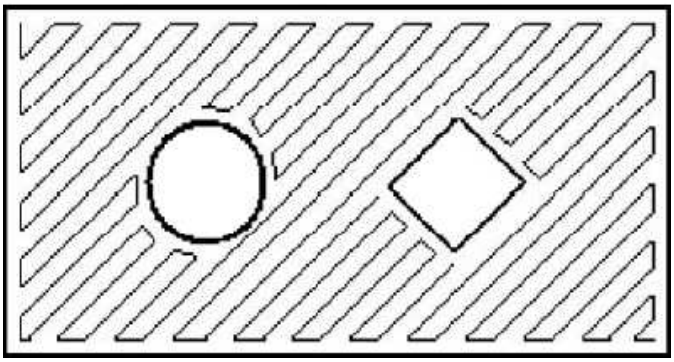


Figure 2.8: Raster example, [27]

This strategy comprises a continuous deposition movement as the tool covers all the surface to fabricate without stopping.

Being one of the most used strategies the literature present a more or less of a consensus about the characterization of this filling method [28, 29, 30, 25, 31]. This is made through the configuration of the following parameters.

Bead/road width is the thickness of the road, which the FDM nozzle deposits or the laser cures. The value of this parameter has to be chosen bearing in mind that, wide roads produce lower building times and overall increase of component stiffness. As a tradeoff the dimensional accuracy is worse and some thin areas may not be filled.

Raster angle is the angle between two consecutive layers. To improve the overall filling, turns of 90 degrees are recommended [29]. This is depicted in figure 2.10.

Air gap is the space between two parallel neighboring roads. A positive gap means that there is unfilled space as the roads do not touch, leading to a more sparse filling that builds faster but produces weaker objects. A negative gap makes the roads to occupy partially the same space, being overlapped. This produces a dense structure that requires a longer time to be fabricated and can degrade the surface quality.

The characteristics desired on the model will affect the choice of parameters. These are shown in the figure 2.9.

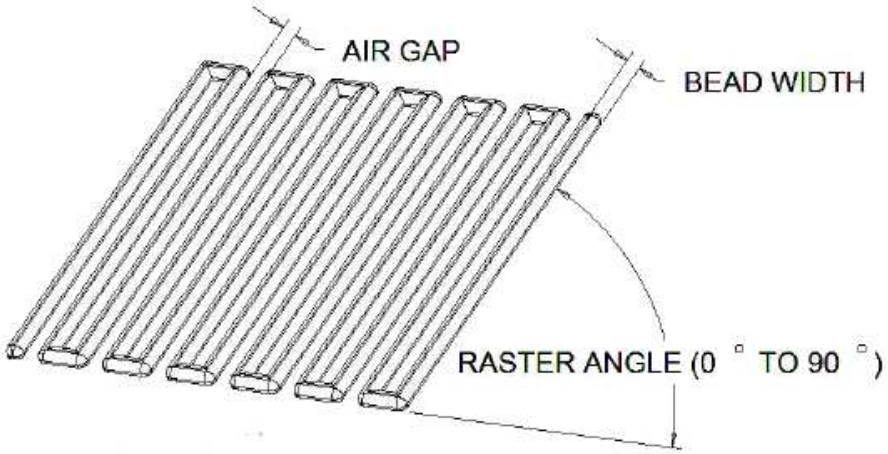


Figure 2.9: Hatching parameters, [29]

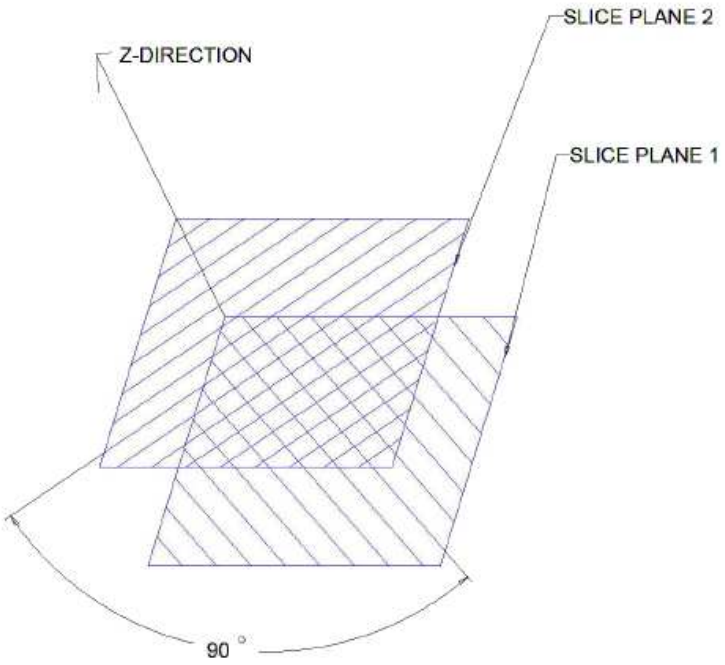


Figure 2.10: Raster rotation, [29]

For droplet printers a hatching method is proposed in [20]. The position of the drops is derived over the raster lines. This can be seen in figure 2.11.

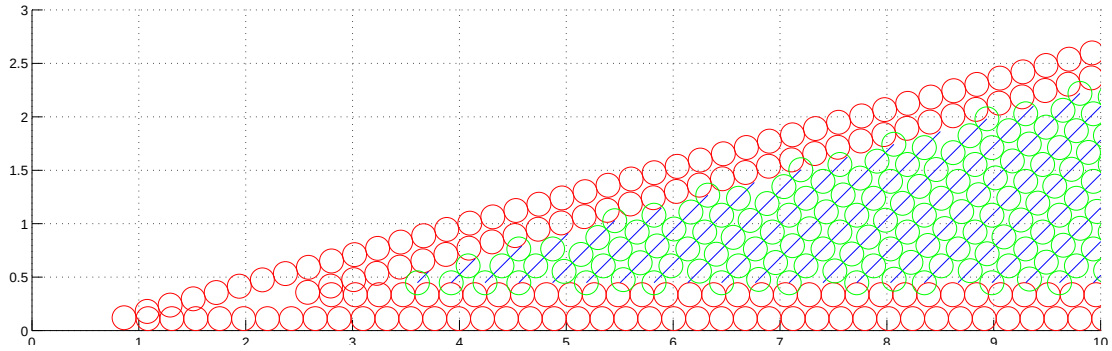


Figure 2.11: Hatching filling example, simulated with MATLAB

This approach allows the use of overlapping with the contours of the slice to prevent the appearance of voids. This overlap is measured in percentage, following a linear distribution: two drops in tangential contact are overlapped 0% and two drops that share the same coordinates are overlapped 100%. The variable in the function is d, the distance between the drops centers.

$$O[\%] = 100 \cdot \left(1 - \frac{d}{2 \cdot r}\right) \tag{2.1}$$

$$d \in [0, 2 \cdot r] \tag{2.2}$$

$$O \in [0, 100] \tag{2.3}$$

Contours filling

Contour filling involves a traversing motion across the curves that including the outer perimeter of the slices are symmetrical but offset to the inside, each curve smaller than the previous until the slice is completely filled. This movement is not continuous as the tool has to move between offset levels to fabricate the next curve [24]. The same model of the previous section can be filled with this method in figure 2.12.

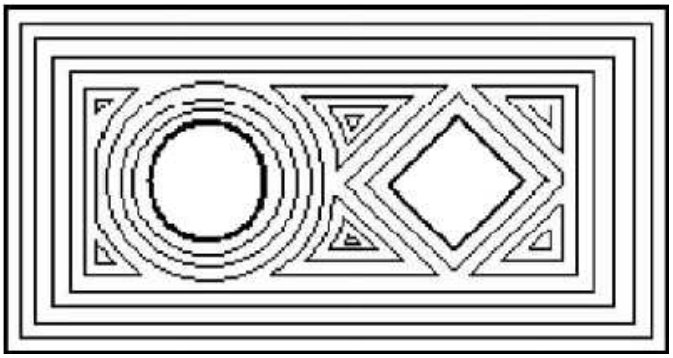


Figure 2.12: Contour example, [27]

If the object has inner holes or a certain geometry that causes these offset curves to intersect, geometrical reasoning is necessary to derive the real filling curves. The computation of these offset curves is covered in different ways in the literature: a NURBS approach is offered in [21], an equidistant path generation method is proposed in [32] and an improved algorithm is presented in [19].

This last algorithm was developed in [20], where there is also presented a droplet contour filling method. In this case the contours are filled with drops. Therefore the contours are offset the length of the diameter of the drops. There are two kind of contours calculated in the slicing, inner and outer, as the part may have inner holes. The algorithm shifts continually the outer contours to the interior and the inner contours to the exterior until all the slice is covered. The contours are constituted of linear segments that form a closed contour and do not self-intersect or intersect with other contours. Once all the contours are calculated the position of the drops is derived over the resulting segments. The result can be seen in figure 2.13.

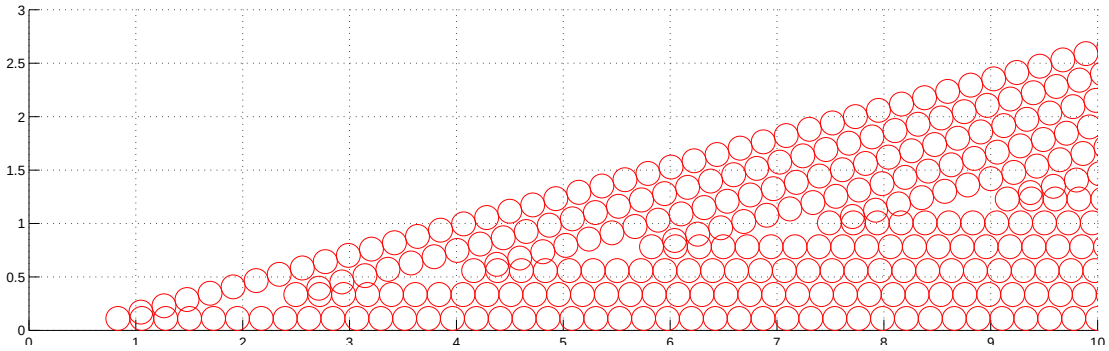


Figure 2.13: Contour filling example, simulated with MATLAB

Other strategies

Based in diverse criteria other filling methods are proposed in the literature. To allow a non-stop fabrication a spiral curve filling (Figure 2.14) is developed in [24], again the spiral approach is mentioned in [22, 23].

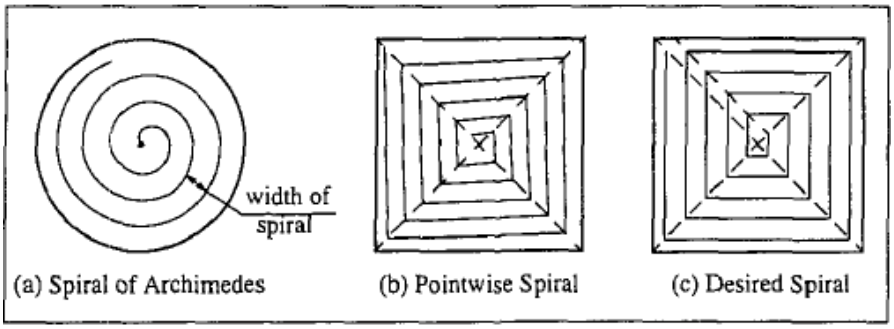


Figure 2.14: Spiral approach, [24]

Looking forward to a better mechanical characteristics a domain decomposition approach (Figure 2.15) is available in [33]. Other special approaches like the use of fractal curves (Figure 2.16) are proposed in [22] and [33].

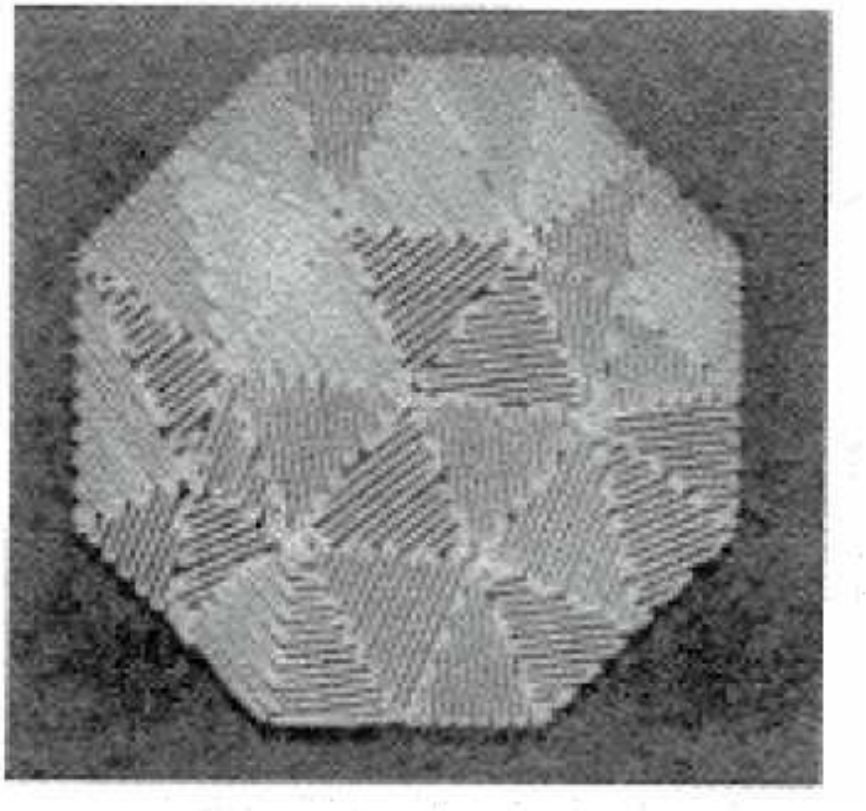


Figure 2.15: Domain decomposition through triangulation, [33]

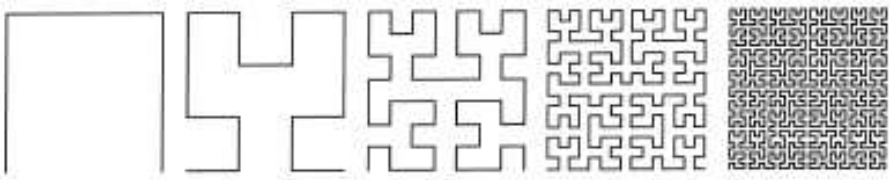


Figure 2.16: Hilbert curves, [33]

For droplet printing one different method is exposed in [34]. The so called dense or grid filling is an implementation in the working environment and new to the field of FDM. The "digital" characteristics of the drop deposition process allows the drops to be disposed independently in different manners.

Considering the spherical form of the drops, the task of achieving the densest filling of the slices turns out to be the sphere-packing problem. This problem will be presented later on.

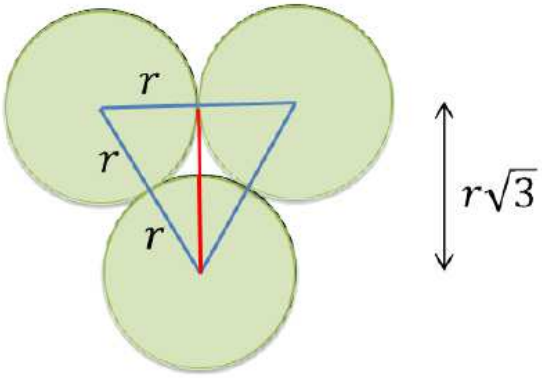


Figure 2.17: Densest circle formation, [34]

This approach is based on the densest circle/sphere formation, represented in figure 2.17. The goal is to fill the most area as possible by avoiding the voids between the drops in formation. To achieve this a raster is deployed similarly to the hatching filling. In this case the drops are shifted between lines to attain the densest formation. An example is shown in figure 2.18.

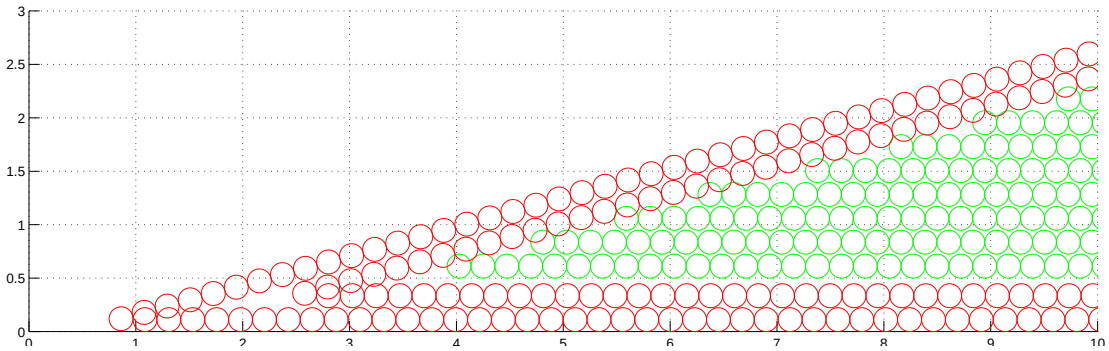


Figure 2.18: Dense filling example, simulated with MATLAB

A more extensive presentation of this innovation can be found on [34].

2.3.3 Strategy comparison

Now that the different strategies available have been presented further comparison is required with the purpose of selecting the best strategy for each case.

For the hatching case, the majority of the paths that the tool follow are straight, allowing it to move faster and therefore reducing building times. As previously mentioned, to improve the filling result the raster orientation is rotated 90 degrees

between each layer. The main flaw of this method is the dimensional and geometrical inaccuracy [22]. As the raster is the same for the whole slices, there may be areas that does not fit sufficiently well and as a consequence will be poorly filled if not even filled at all.

The contour approach produces the best surface quality as it exactly follows the perimeters of the slice. In this case, the arbitrary geometry may appear in form of complex curves that the tool must negotiate in a somewhat slow motion to achieve the aforementioned surface quality. This implies increased building times [22].

The other strategies proposed, as the spiral curves or the fractal curves, are very limited in application, since they cannot deal with arbitrary geometries and taking the complexity of the fractal curves it may turn out in too large building times.

As a solution that takes the advantages of the two most used approaches, a mix of these both is found extensively in literature [22, 35, 25, 36]. The idea is to use a contour tool-path along the boundaries to obtain the desired surface quality and then complete the inside with a hatching filling improving efficiency and reducing the time required.

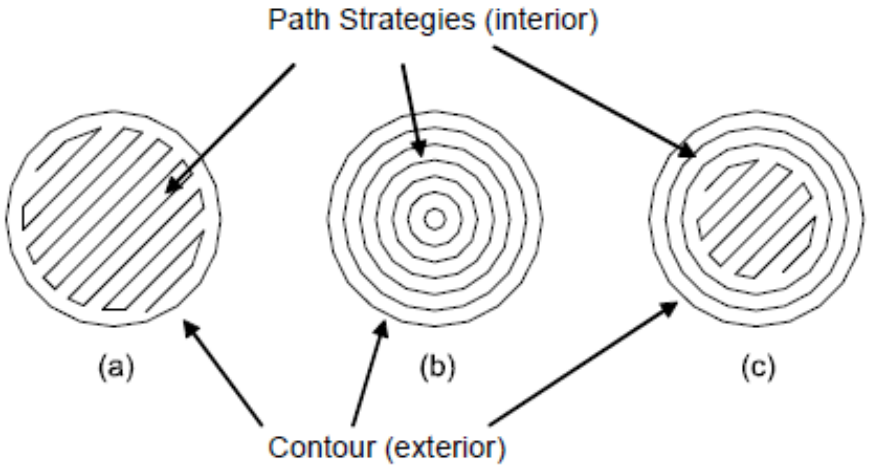


Figure 2.19: (a) hatching, (b) contour-only and (c) mixed solution, [36]

2.4 Packing problem

Understanding the geometrical functioning of the printer used, that the printed unit is but a sphere, and that we want to optimize the filling, becomes natural to think about the proposed solutions in the literature about the traditional problem of the packing of spheres.

In [17] the problem is defined as follows:

Place $n \geq 2$ identical circles (disks) in a given, bounded subset of the plane without overlapping, in such a way that the density of the packing is maximal. (The density of the packing is defined as the ratio of the filled and total available area.)

The application of this problem to our printing case would be to find the densest filling based in spheres for a certain arbitrary geometry. To make it viable new conditions would be needed, because the problem, as it is, does not consider surface quality or geometrical tolerance.

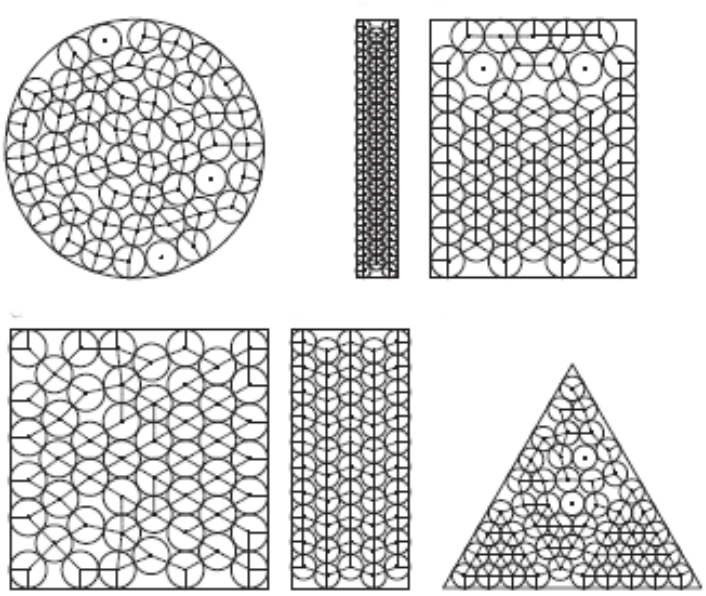


Figure 2.20: Sphere packing examples, [37]

In literature different models are proposed: non-linear models [37], linear models [38] and optimization approaches [39].

The advantage of this idea seems to be the high density achieved in the filling. On the other hand the models proposed in literature involve a high time consumption when the geometrical complexity and size of the boundaries increments [37, 38].

Some examples of solutions are presented in figure 2.20.

RP is a tool known for offering models and prototypes in a short period of time, being speed an important characteristic of this kind of processes. It is undesirable then that one of the phases of the process slows down the whole procedure.

Chapter 3

Limitations of the State of the Art

In this section the limitations and issues present in the actual approaches of layer filling will be presented. In the literature used for the state of the art research in this field some limitations and defects are stated. These issues are the "staircase effect" and the defects caused by the various filling methods. In some cases solutions of improvements are proposed for these limitations.

3.1 Staircase effect

This error can be defined as the inaccuracy or the discontinuities [40] produced by the approximation of curved surfaces by discrete steps [29] of layer thickness. In [25] is stated that this is a common problem, to some extent, to all layer manufacturing processes. This can be easily understood by the need to approximate curved surfaces (Figure 3.1) through layers of material that always leave some space unfilled, since there is no possibility to perfectly adapt rectangular step to a curve.

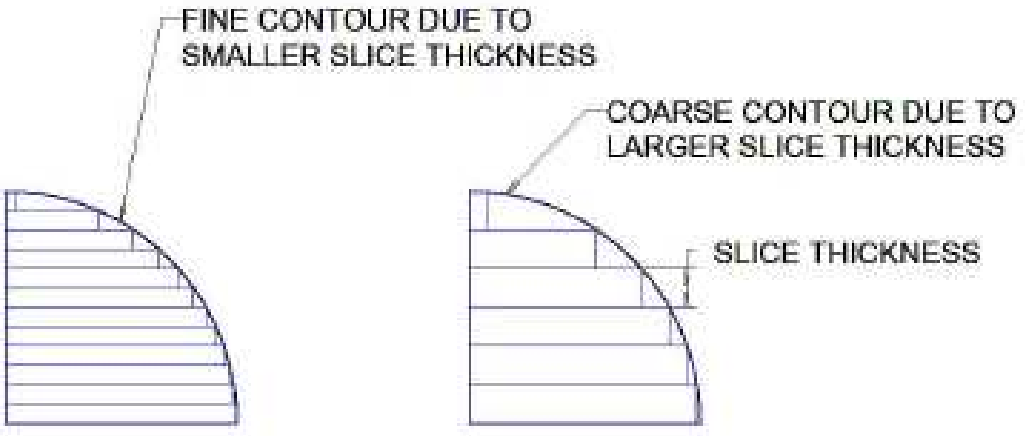


Figure 3.1: Slice thickness effect, [29]

In practice some decision has to be made considering surface quality or building time. The thicker the layer, the faster the model will be built, at the expense of reduced surface quality. In [22] a tolerance distribution is defined in three scenarios, being the decision here to accept the exceeding of boundaries or not.

Another approach to deal with this limitation is to implement an adaptive solution for layer thickness derivation. If there is the possibility to vary the layer thickness along the model, a better surface quality and geometrical accuracy can be achieved (Figure 3.2). This is not the case of every LM process

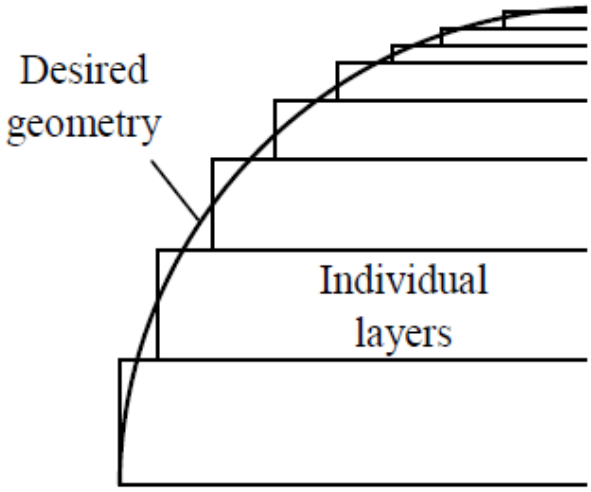


Figure 3.2: Adaptive layer thickness, [40]

3.2 Raster Filling Defects

The raster filling is by far the most used in practice, for that reason its limitations and the defects that it causes have been comprehensively studied in the literature. In this subsection two kind of defect will be distinguished, the actual in-layer voids and the possible overfills.

3.2.1 Voids

As in [22, 24], the void is defined as the area not filled by the process, due to the inaccuracy of the system. The use of the raster filling method in certain geometries comprises a variety of voids.

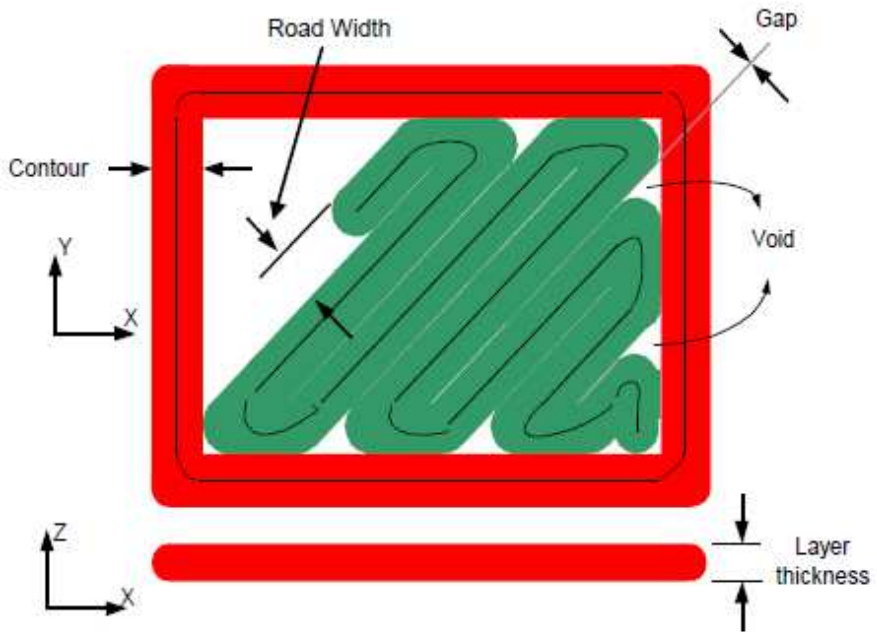


Figure 3.3: Void Error, [36]

In [25, 35] these voids are classified in two groups: sub-perimeter voids and inter-road voids (Figure 3.4). The sub-perimeter voids appear on the area inside the perimeter of the object as the road turns tangent to the perimeter leaving space without filling. The inter-road voids can be caused by mechanical reasons, as slippage of the feed mechanism or inconsistent material properties, or by discontinuities produced in convex areas.

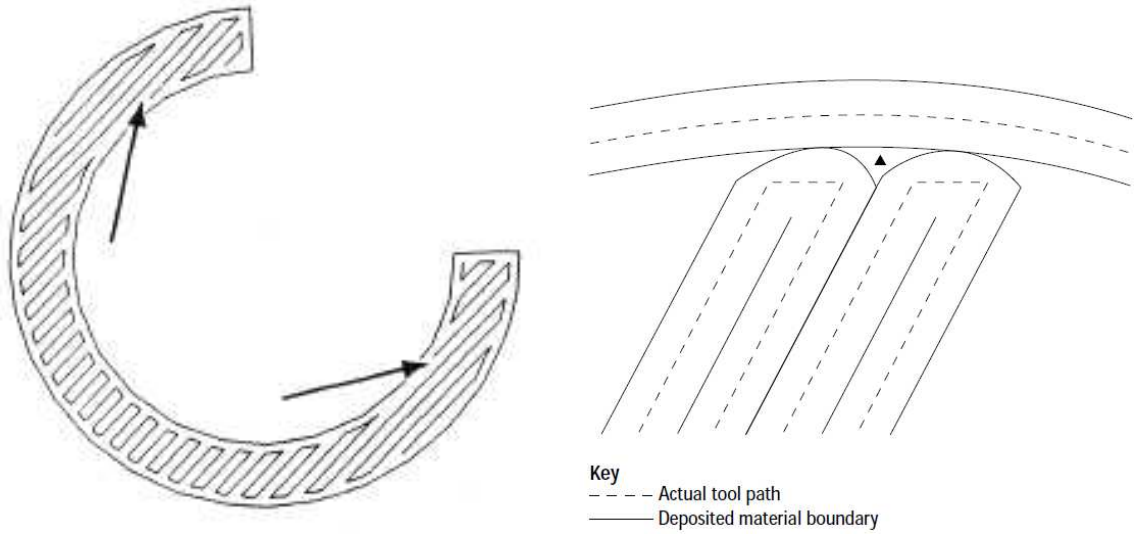


Figure 3.4: Interroad [35] and Subperimeter Voids [25]

Another classification is proposed in [41]. There are three kinds of void existing on layers of part: sub-perimeter vector-direction void (SVV), sub-perimeter road-turn void (SRV) and irregular void (IV). SVV is sub-perimeter and on the vector direction. SRV is sub-perimeter and between two adjacent roads turns. IV is because of irregular behaviors of machine or material flow. This classification is depicted in figure 3.5.

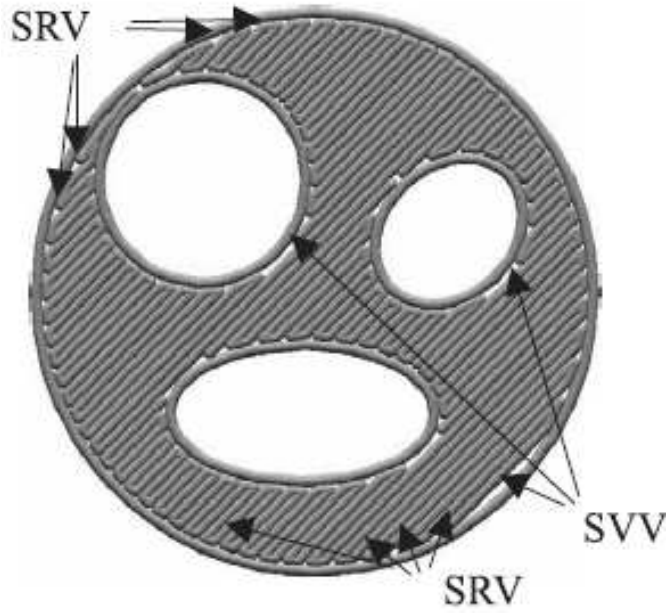


Figure 3.5: Void types, [41]

The sub-perimeter voids, in both classifications, are minimal whenever the angle of the raster with the boundary is as near as possible to 90 degrees.

For these both classifications of voids there are some proposed solutions (Figure 3.6) to repair or avoid the generation of voids. They consist typically on causing an overlapping of material to force the filling of the untouched areas. Be that with negative perimeter offset [35, 25] or modifying the road width, not possible in every machine, as it goes through critical areas [41].

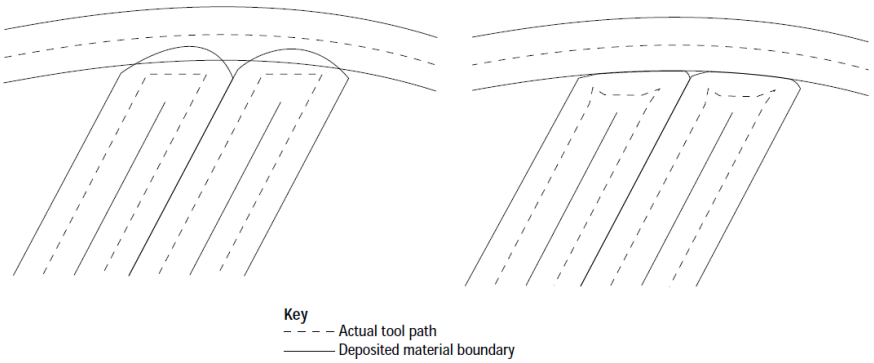


Figure 3.6: Solutions proposed, [25]

3.3 Overfills

The characteristics of the continuous deposition or curing of material involve a limitation that should be taken into consideration. As the tool follows the path, if it turns too close it may go over recently deposed or cured material, producing what is defined as an overfill. These overfills usually entail also underfills or voids [26]. Trying to avoid these overfill cases usually turns out to generate more voids of unfilled areas that only would be filled by overfilling others.

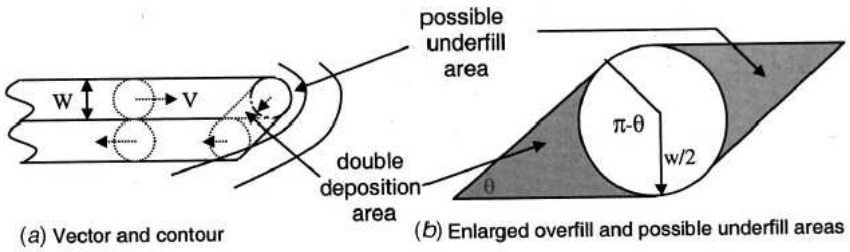


Figure 3.7: Over- and Underfilling case, [26]

To improve this situation an adaptive tool-path approach could be an intermediate solution to avoid these kind of path that comprise overfillings.

3.4 Contour Filling Defects

Being a less used filling method, there is not much research done about the void appearance. For this reason a graphical study was conducted in this thesis connecting with the research already available in the literature.

The voids will be classified in four classes: core void, outer-inner boundaries encounter, shape change void and acute corner case.

3.4.1 Core void

As offset contours are calculated and get deeper to the center of the slice, in one moment there will not be enough space to fit one more contour and a void will be produced if the road width is not exactly a multiple of the part dimension [35, 25].

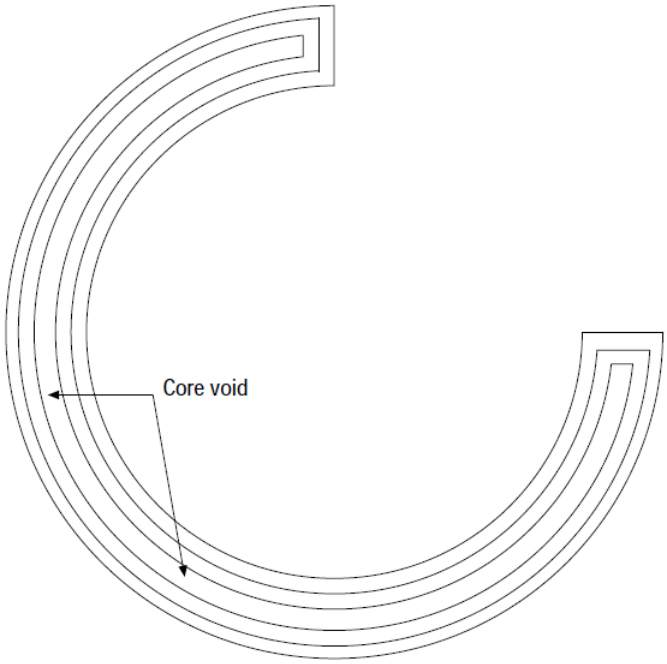


Figure 3.8: Core Void, [25]

For this situation a calculation for the optimal road width is proposed [35, 25], but again this solution is not viable for all kind of machines.

3.4.2 Outer-inner boundaries encounter

Pieces with inner holes in the geometry are prone to produce inhomogeneity when the contour offsets from the outer and inner boundaries meet, causing voids to appear, for example in figure 3.9.

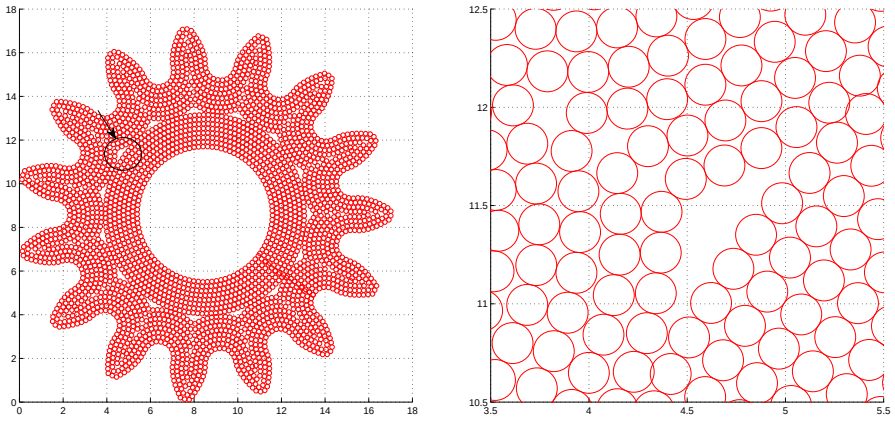


Figure 3.9: Part filled with droplet contours, simulated in MATLAB

In the figure, the encounter of offsets can be observed and, as this contours are not from the same boundary, their shape do not produce a desired smooth encounter, causing unfilled voids to appear.

3.4.3 Shape change

Pieces with certain geometry will produce offset contours that change their shape as they progress deeper in the slice. This change of shape will generate an uneven encounter between contours, causing voids between these contours.

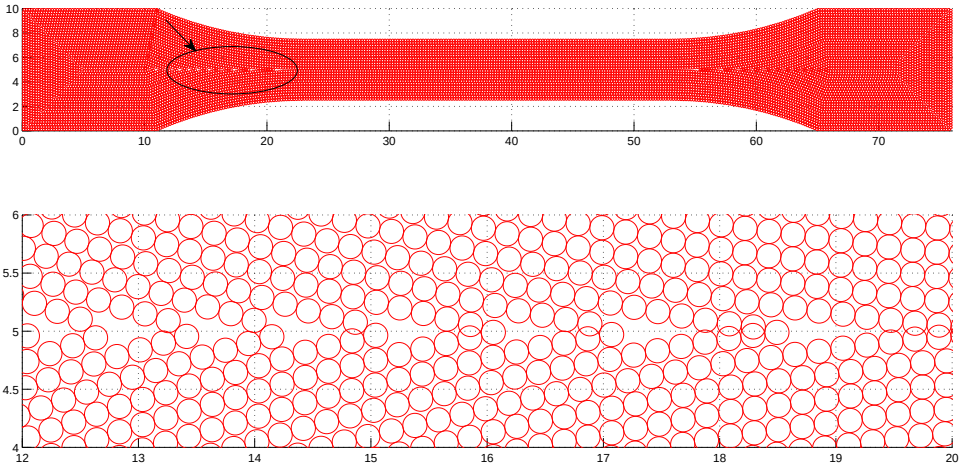


Figure 3.10: Contour shape change in a droplet part, simulated in MATLAB

3.4.4 Acute corner void

In [40] the physical limitations of using a FDM nozzle with fixed radius lead to a constraint on the angle of the corners that can be correctly printed. Considering the nozzle a perfect circle, corners with an angle acuter than 60 degrees will not be fully printed, leaving a void. This is a real limitation as acute corners cannot be printed without overlap.

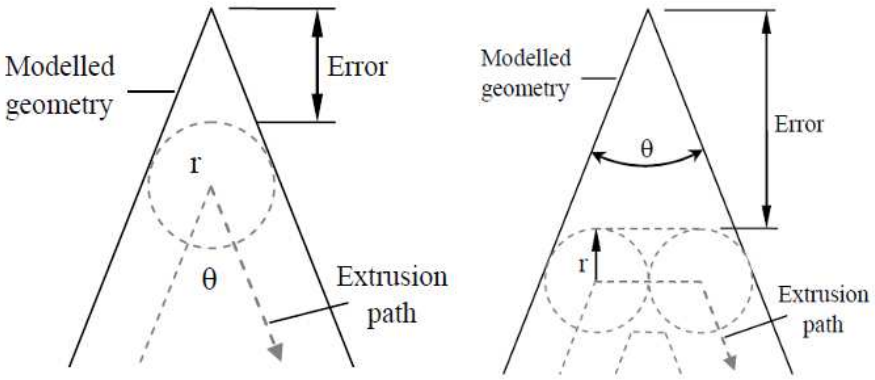


Figure 3.11: Corner effect, [40]

In the figure 3.11 can be recognized that trying to fill the most of the contour results in overlapping and if the overlapping is to be avoided the resulting error is even bigger. At the moment the solution depends on the availability of smaller radiuses of nozzles. Also if the corners are acute enough the successive contours will propagate the void generated as they resemble the corner shape of the boundary.

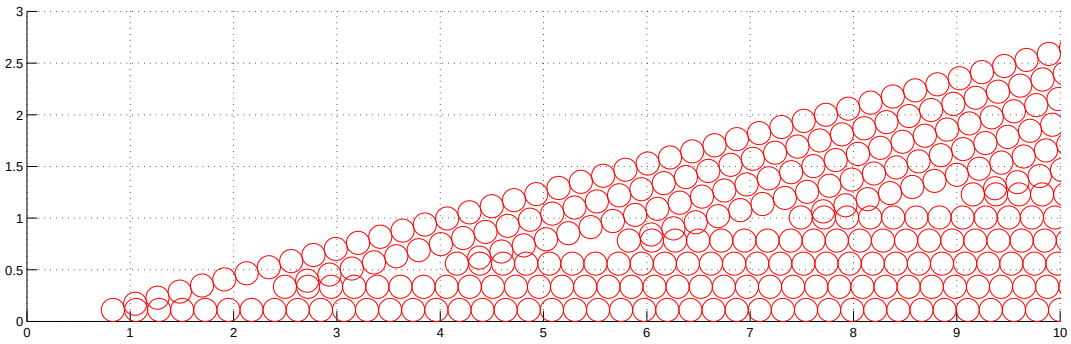


Figure 3.12: Acute corner void case, simulated in MATLAB

3.5 Grid Filling Defects

Some parts were simulated (Figure 3.13) using the dense/grid filling method to observe the performance. Voids were detected and therefore the statement that this approach is not directly acceptable was proposed.

The dense filling is constrained by the same condition that makes it the densest. That is the geometrical disposition of the circles that produce angles multiples of 60 degrees. Whenever the perimeter meets the grid in a way that the lines are not parallel, voids will appear to avoid overlapping. Also when the dimensions of the part do not allow the filling to be complete as the grid cannot be fitted perfectly.

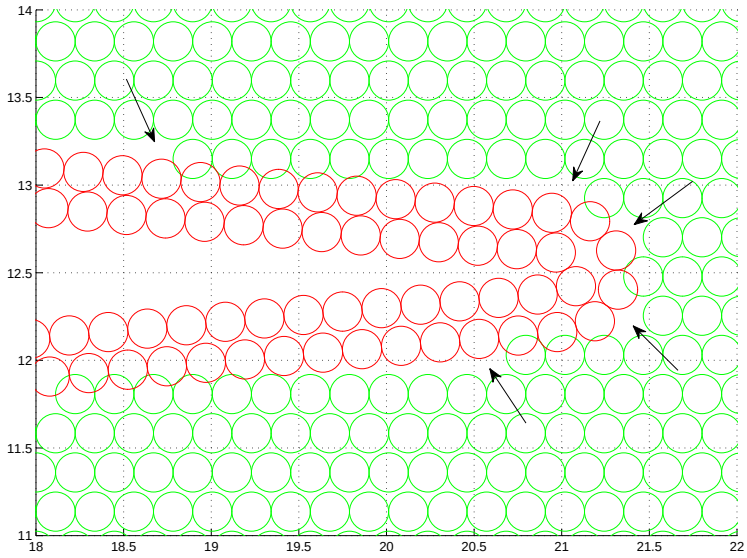


Figure 3.13: Dense filling void cases marked with arrows, simulated in MATLAB

Part II

Own Approach

Chapter 4

Task

The primary objective of this thesis is to create a tool that detects and fills the voids that may appear in a previously calculated tool-path of the slice filling phase in a certain AM process.

As seen in the previous chapter, the actual approaches on slice filling have limitations and given the physical characteristics of the machines, the optimal tool-path layouts tend to leave voids. Therefore this new tool is intended to solve this situation, improving the tool-path by adapting the infills to certain geometries, which are not always the same, as they are treated in an individualized manner.

After obtaining a new enhanced tool-path, some kind of test is deemed indispensable to evaluate the results of the implementation. To accomplish that, two experiments will be carried out. A software test will be specially implemented to assess the newly generated file, as for amount of new material volume added and the overlapping consequently produced. Subsequently a microscopical evaluation will be performed over parts printed with the new implementation to observe the elimination of the aforementioned voids.

The goal criteria will be then to obtain positive results in the difference of new volume added versus overlapping produced. For the microscope test, the comparison between normal parts and parts with new algorithm should give a positive result, regarding a drop on the size of the voids encountered.

Chapter 5

Expected Functions and Benefits

When configuring the slice filling in the AM process sometimes a sparse filling is preferred and the voids in this cases are of no interest. But when the goal is to obtain the densest filling possible or that the mechanical application of the part has high strength specifications these voids may be critical and their filling would avoid unacceptable failures.

The stress concentration effect is defined in [42] as: *The magnification of applied stress in the vicinity of a notch, hole, inclusion, or inside corner. Minimizing stress concentrations is an important aspect of plastics product design*

In figure 5.1 an ideal representation of the concept can be observed.

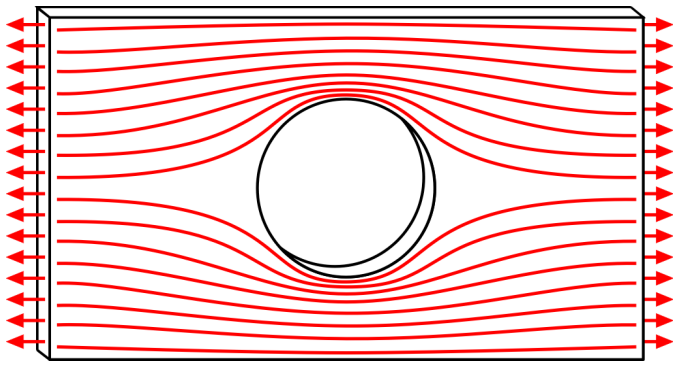


Figure 5.1: Internal force lines are denser near the hole, [13]

Considering this, if the appearance of holes can be avoided, an improvement on the part stiffness and mechanical capabilities is to be expected. Besides the mechanical aspect of the part, also the surface quality and the visual quality are estimated to improve as those holes and voids will not be noticeable by the user.

The main function of this implementation is to prove the suitability of including such hole filling idea in an available software workflow for the slice filling phase.

Following the configurability available for the common slice filling processes, in which the overlapping factor for the filling is usually adjustable, in this implementation that parameter will also be modifiable.

As presented in the literature research, for filament deposition some work on this field has been performed [41]. In that paper an approach of detection was proposed with the requirement of changing the physical characteristics of the deposition, as may be the dimensions of the deposited filament. This requirement may be difficult to fulfill, therefore some other approach is needed.

Chapter 6

Hole Detection Problem

One of the key tasks is to detect the existence of holes and their position in the plane. The first step is to return to the literature to see how this problem is dealt with. To perform this research deep knowledge of the environment is required. The printer in use works with droplets, which in the data file are nothing else than a point with 3 coordinates. Therefore they can be treated as a cloud of points or circles.

The criteria for this research comprises point set/cloud analysis, hole detection and point set boundaries. Three papers were found on the fields of computer graphics and communication networks. They are not related with AM or LM, but the mathematical base that they share is very similar.

In [43] different criteria for boundary localization is presented. This boundary detection is offered as useful for many different applications. For the idea of hole detection, one of the proposed criteria is the angle criterion.

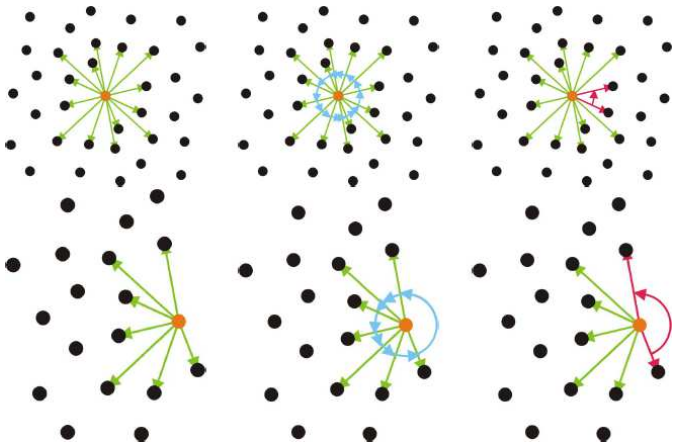


Figure 6.1: Angle Criterion, [43]

The angle criterion works as follows: once calculated the direction vectors of all the nearest neighbors at some distance of the center, the angles bigger than some threshold will indicate the presence of a boundary or hole. This process is presented in figure 6.1.

The problem of holes in sensor networks is often treated in the literature. In [44] another nearest neighbors (NN) approach is presented. The idea is to calculate the NN of the chosen point. Those NN are then taken as points and their own NN are calculated. Then the NN groups are compared: those points that share less neighbors with their neighbors are to be considered candidates of being in the boundary.

Lastly a different mathematical tool was encountered in [45]. In this paper the authors focus on detecting coverage holes in sensor networks. For that, they apply an algorithm based on Voronoi diagrams. This mathematical construction has interesting properties for the problem in hand and will be presented later on in the next chapter.

Chapter 7

Structure Description

The product developed in this thesis is a software implementation, but, as any machine or apparatus, it has also various modules that execute different tasks. These tasks produce distinct outputs that end in the desired goal. Hence the program can be divided in four modules:

1. Parser
2. Analyzer
3. Hole filling algorithm
4. GCode writer

The implementation will be done in MATLAB. The input is generated through an available C++ program.

7.1 Parser

The parser analyses the formatted text ".gco" input file that contains all the orders for the machine to print the part. Changes in the format will require an update of this module. These orders can be simulated to reproduce the part that is going to be printed and observe the final result. This simulated orders result in the position and features of the drops being printed. It also reads the headers of the file that contain the main information of the part, such as filling style, slice height and number of contours being printed. The output generated is the position and characteristics of the drops.

7.2 Analyzer

Taking as input the output of the, the analyzer sorts the drops in two steps. First according to their z-coordinate they are arranged in slices. Secondly, given their features they are classified in contours or filling. This idea is shown in figure 7.1. The output produced is the organized drop data.

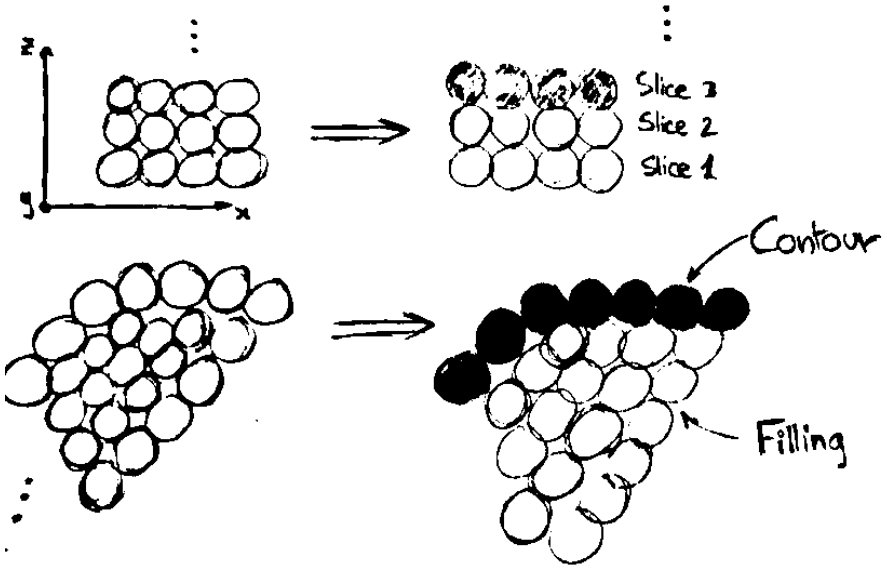


Figure 7.1: Analyzing Processes

7.3 Hole Filling Algorithm

Provided a set of analyzed drops as input for the algorithm, it works through 2 steps. First, it detects the possible voids that the drops may form. In the second step, if the overlapping thresholds hold, the detected void is filled with new drops. In figure 7.2 this procedure is depicted.

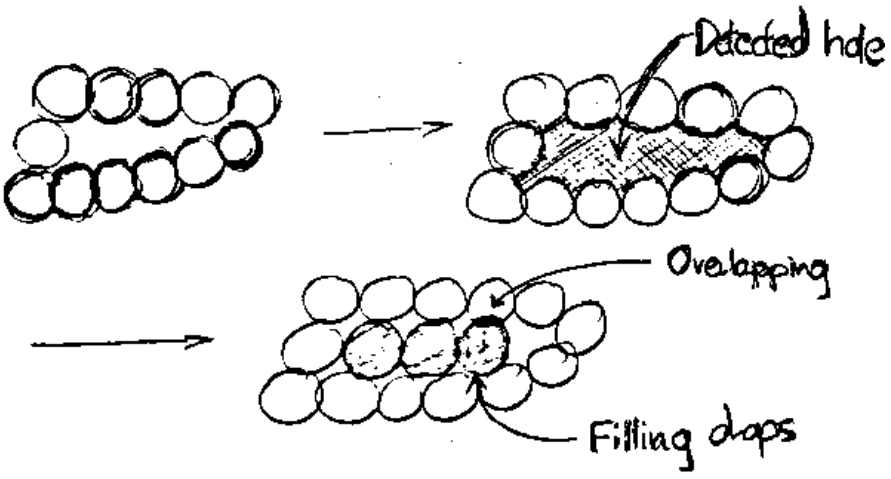


Figure 7.2: Filling Process

Through this thesis two algorithms were tested, one based on a nearest neighbors (NN) approach and another using Voronoi diagrams. The comparison of both algorithms is presented in the next section of this chapter.

7.4 GCode Writer

This last module produces the new .gco file with all the old drops plus the new void filling drops. The format should be the same that of the input file of the program. An ideal example is presented in figure 7.3.

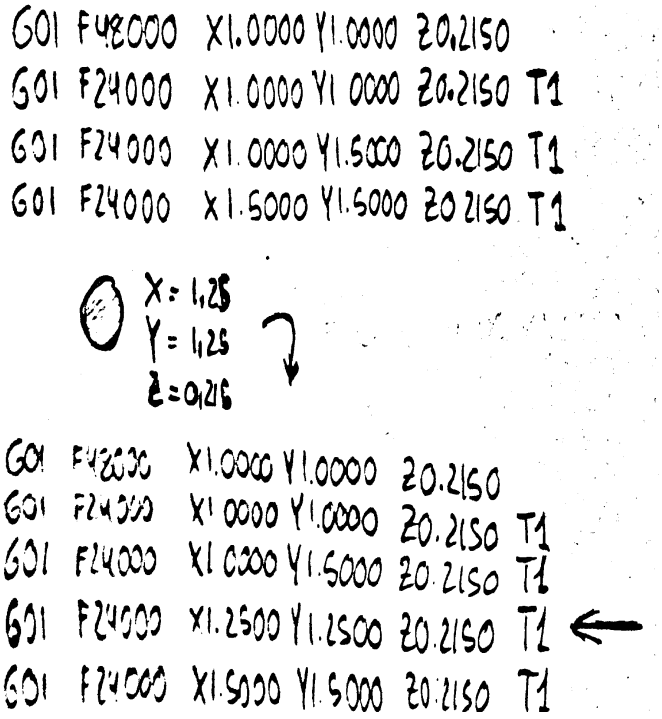


Figure 7.3: GCode generation

Chapter 8

Process Description

In this section the whole process from the "initial GCode" file to the "new GCode" file will be presented through flow charts and the deemed explanation. Once the general process is presented, the processes on composing modules will be described.

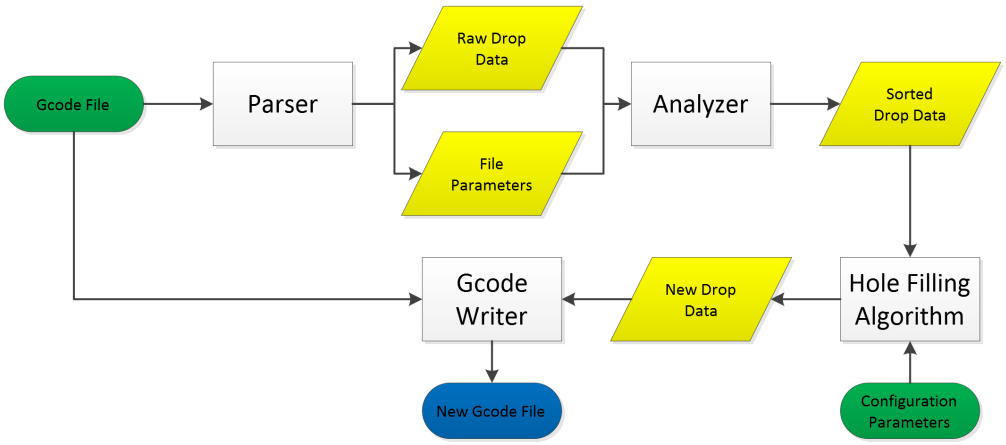


Figure 8.1: General Workflow Chart

The two main inputs of this implementation are the already mentioned GCode file and the configuration parameters. These parameters are user-selected and comprise the choices of execution, meaning what parts of the application are to be executed and most importantly, the overlapping threshold, which will be depicted later.

The following subsections will show the processes of each part of the presented general process.

8.1 Parsing

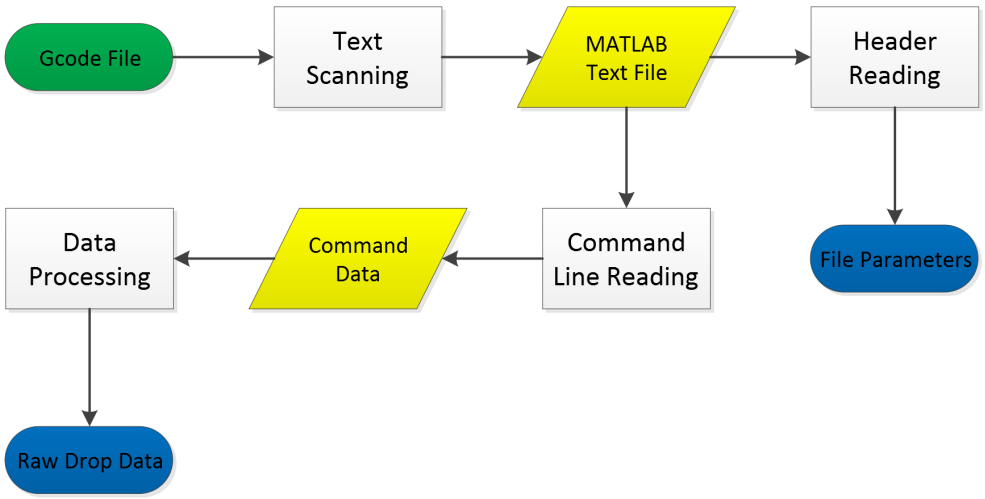


Figure 8.2: Parser Workflow Chart

The format of the GCode file is critical on this phase, as it determines the way the text is going to be read. Different formats can be accepted at the expense of increasing the program complexity.

The commands read are the ones already presented on the introduction. The file parameters read contained on the headings of the file are: drop properties, slice height, order of contour printing and order of filling. The drop properties refers to the swelling produced on the drop after deposition. The orders of printing apply to the sequence in which the drops are printed. It can be chosen that the inner filling is printed after or before the contours or that the inner contours will be printed before the outer ones.

The finally extracted and processed data is the list of all coordinates of each command and the corresponding command feature. The file parameters will be required by following phases.

8.2 Analyzing

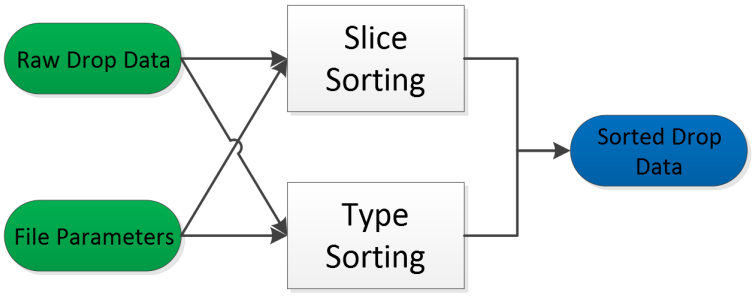


Figure 8.3: Analyzer Workflow Chart

As mentioned before, the analyzer sorts drops in slice and type. With the slice height, the number of slices is calculated and those drops with the same coordinates are grouped in the same slice. Now for each slice and according to the printing order the drops are divided in contours and filling.

8.3 Hole Filling

This is the core part of the program, since the existence of voids will be tested and if necessary, the additional filling will be executed.

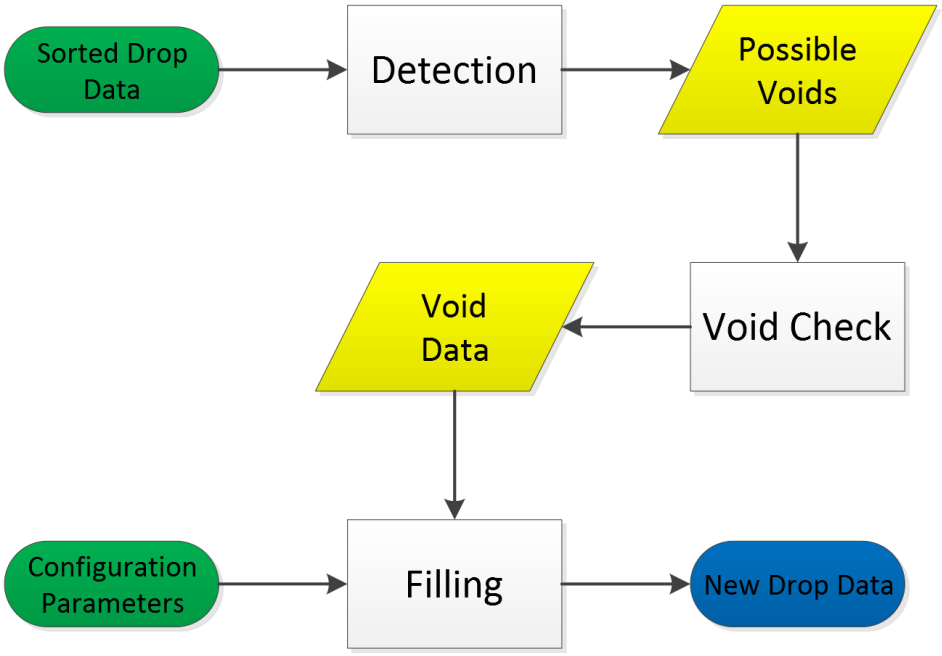


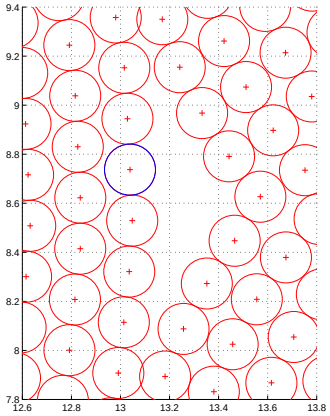
Figure 8.4: Hole Filling Process Workflow Chart

8.3.1 Detection Process

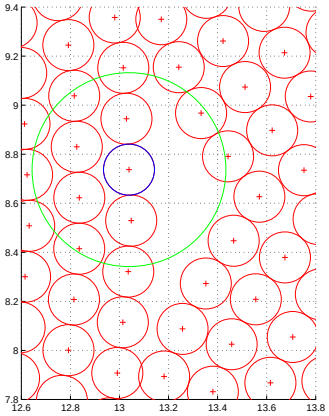
For the detection part two approaches were proposed: a nearest neighbor (NN) procedure and an application of the Voronoi diagrams. In the following subsections both ideas will be depicted and compared.

Nearest Neighbors Procedure

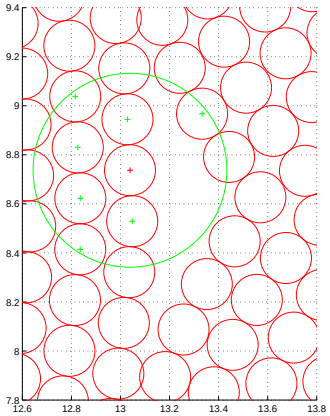
This approach is the same seen in the topic's research. It is represented in figure 8.5.



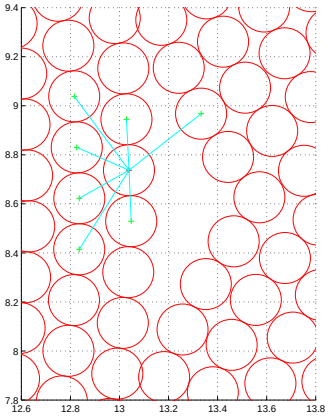
(a) Void to check



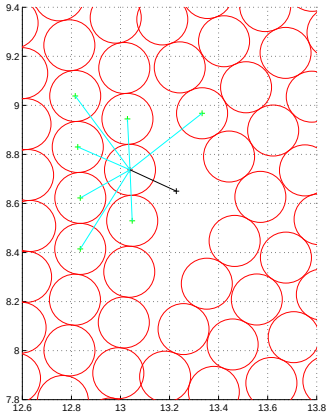
(b) Area of possible neighbors



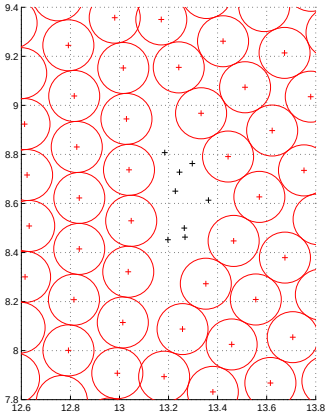
(c) Nearest neighbors found



(d) Vectors to NN



(e) Point calculated in the bisector of the bigger angle



(f) Resulting points of all boundary drops

Figure 8.5: NN Process, simulated in MATLAB

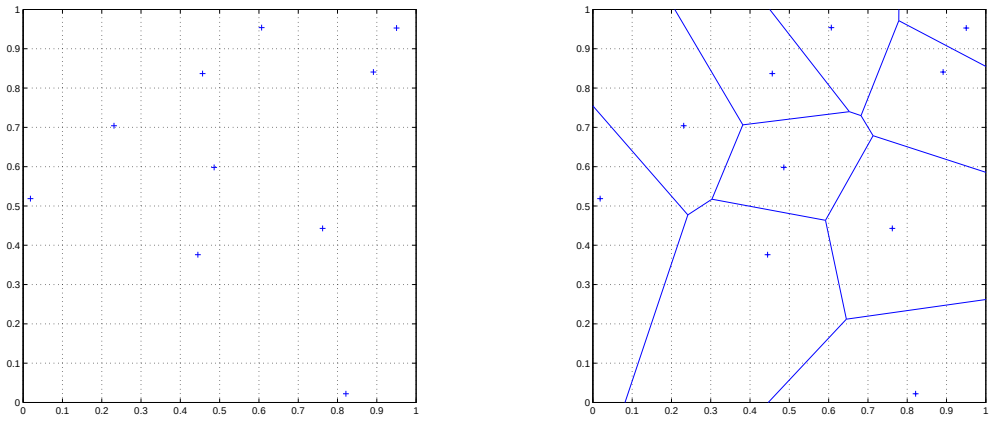
The process functions as follows:

1. For each drop, the nearest neighbors (NN) drops inside a certain distance are calculated (Figure 8.5b and 8.5c).
2. The vectors joining the drop to its NN are calculated (Figure 8.5d).
3. The angles between the adjacent vectors are calculated.
4. The angles bigger than a threshold are considered to be facing a void.
5. A point in the bisector of each of the bigger angles is calculated at a certain distance (Figure 8.5d).

Each drop on the boundary of a void generates points. This clouds of points (Figure 8.5f) mark the existence of a void and will be treated later on.

Voronoi Diagrams Application

As in [46], a Voronoi diagram (Figure 8.6) is: *The partitioning of a plane with n points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other. A Voronoi diagram is sometimes also known as a Dirichlet tessellation. The cells are called Dirichlet regions, Thiessen polytopes, or Voronoi polygons.*



(a) Cloud of points (b) Voronoi Diagram of those points

Figure 8.6: Voronoi Diagram generation in MATLAB

The vertices of the Voronoi polygons will be referred to as Voronoi vertices in this document. These vertices can be extracted from the data of the Voronoi diagram.

The approach, presented in figure 8.7, works as follows:

1. The Voronoi diagram is computed using the drop 2D coordinates as input (Figure 8.7a and 8.7b).

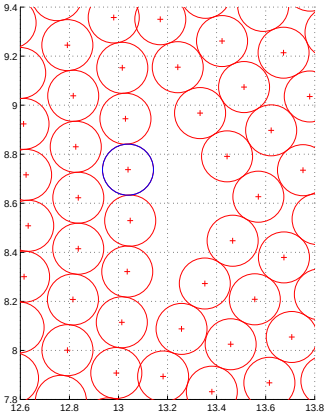
2. The vertices of the diagram are extracted (Figure 8.7c and 8.7d).

3. The distance of the vertices to the nearest drop is calculated.

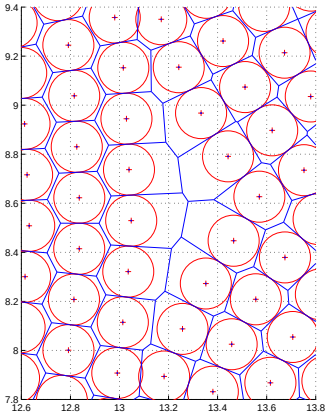
4. The vertices with a distance bigger than a threshold are selected (Figure 8.7e).

Those points selected (Figure 8.7f) will mark a void and will be treated later on.

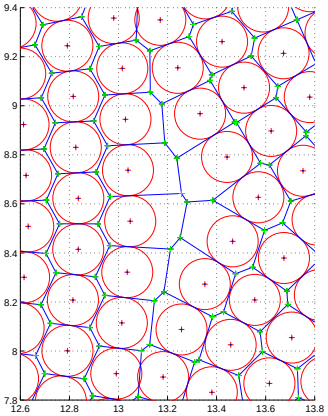
Generating the Voronoi diagram of the drops will give the coordinates of the Voronoi vertices. Whose distance to the drops can be calculated and then those complying with a threshold are chosen.



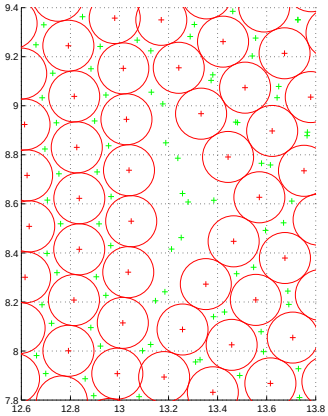
(a) Void to check



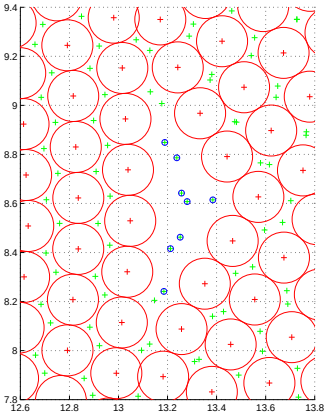
(b) Voronoi Diagram



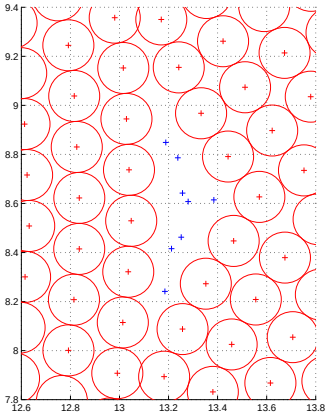
(c) Vertices in the Diagram



(d) Voronoi Vertices



(e) Selected Voronoi Vertices



(f) Point Result

Figure 8.7: Voronoi Process, simulated in MATLAB

Approach Comparison

Before proceeding with further implementation a time study was conducted to, considering that in RP speed is a core criterion, choose between the 2 proposed approaches. With 3 different models, 3 filling methods each, simulations were executed, the time results are shown in the following graph:

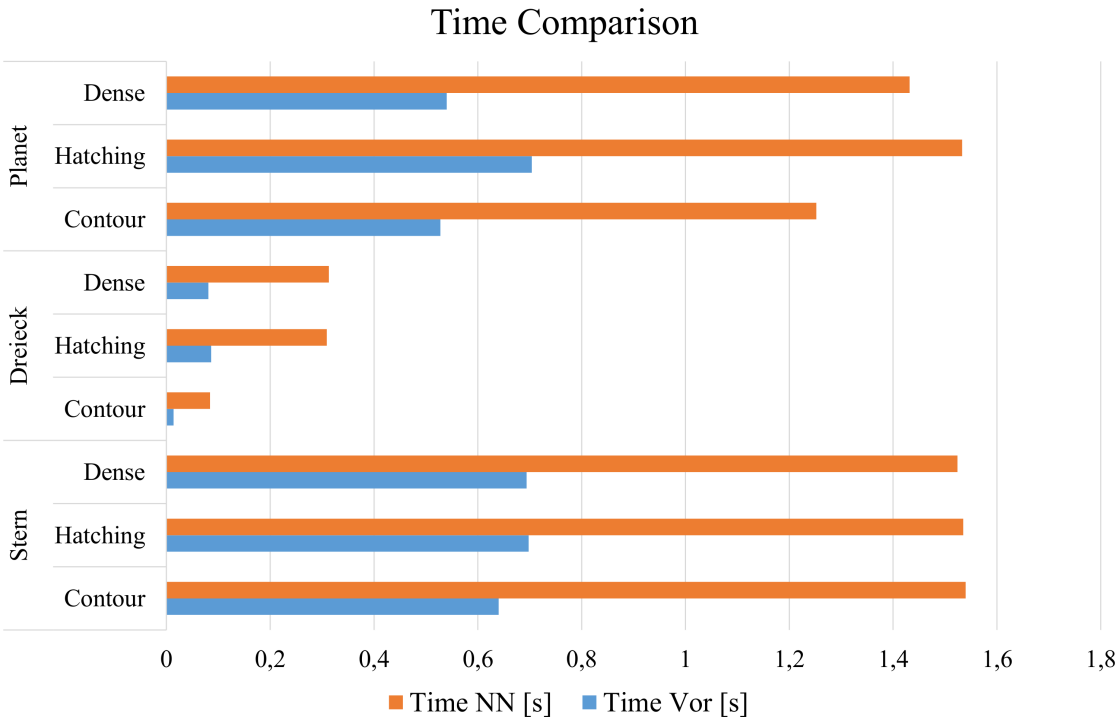


Figure 8.8: Time Test Results

So far can be observed that in every case the Voronoi approach is faster that the NN one. Therefore the chosen method for the main implementation will be the Voronoi diagrams application.

8.3.2 Void Checking

Once all the possible voids have been identified and located by the points inside of them, those points are going to be tested for overlapping. That means that those points that are too near to other drops will be eliminated. To test the overlapping, a threshold is configured as an input parameter. The overlap (O) is taken as a measure of the distance (d) between two drop centers, following a linear function:

$$O(d = 0) = 100\%; \tag{8.1}$$

$$O(d = 2 \cdot r) = 0\%; \tag{8.2}$$

$$O[\%] = 100 \cdot \left(1 - \frac{d}{2 \cdot r}\right) \tag{8.3}$$

When we introduce the threshold O_t all the points with a distance smaller than d_t will not pass the test.

$$O < O_t \rightarrow d > d_t = 2 \cdot r \cdot \left(1 - \frac{O_t}{100}\right) \tag{8.4}$$

8.3.3 Filling process

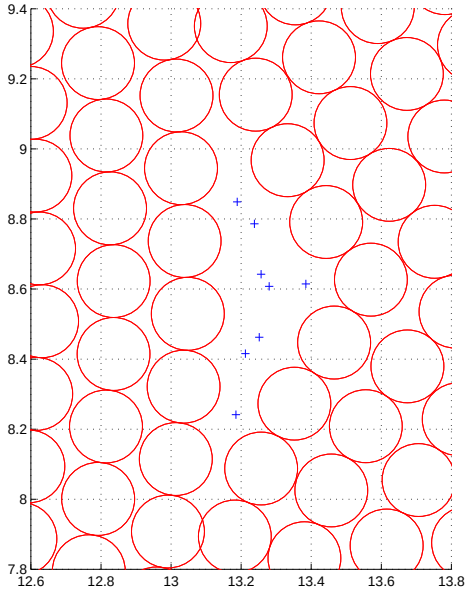
In this phase the points produced in the detection stage are going to be treated to derive a filling solution for the voids. The process is presented in figure 13 and exposed in the following steps:

1. The points of each void are sorted on a list to build a line that joins them. This sorting takes the most far away point from the center of the void as the first one. Then continues adding the nearest point to the previous on the list, till all points are sorted in the list.(Figure 8.9b)

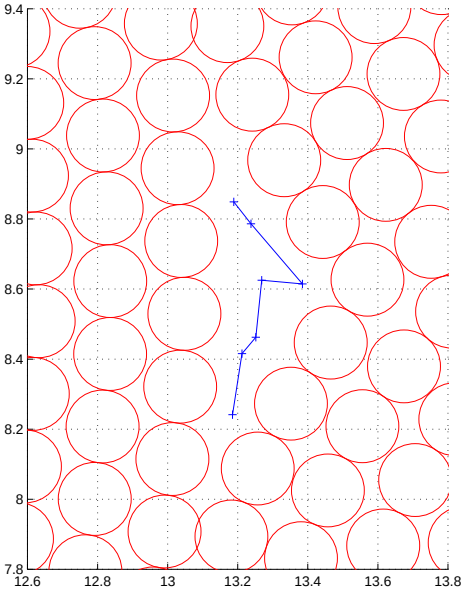
2. The line is filled with drops. The filling put drops whenever they do not overlap within themselves and they do not overlap more than the threshold with the drops of the part. To adapt to this conditions the drop positions are calculated around the line.(Figure 8.9c)

3. The new drops are stored and also the id of the nearest part drop (in red in the figures). This information is needed for the last phase of the program.(Figure 8.9d)

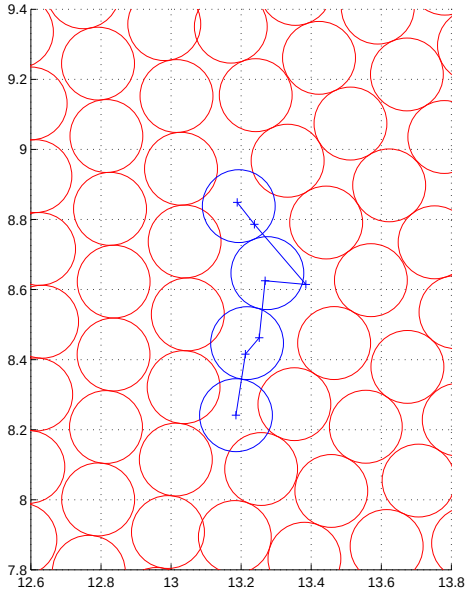
In certain geometries there may be new drops calculated outside from the part. This drops are incorrect and have to be eliminated. The contours of the slices are known, as they are analyzed in the previous phase. With this, it is possible to check whether the drops are inside the contours or not. Those outside will be eliminated.



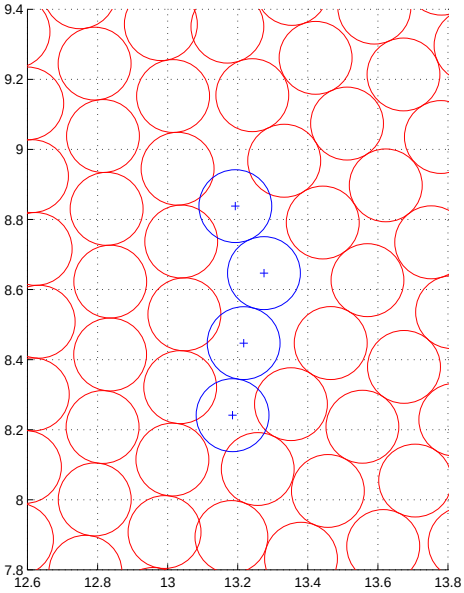
(a) Points in the Void



(b) Void Line



(c) Void Line Filling



(d) Result of Filling

Figure 8.9: Filling Process, simulated in MATLAB

8.4 GCode Writing

The last phase of the implementation is to generate a .gco file with the input data and the new calculated drops that fill the voids. The procedure is as following:

1. The input .gco file is read.
2. The new drops are sorted according to the id of the input drop that was assigned in the previous phase.
3. Whenever an id that is on the new drop list is found, the new GCode lines corresponding to the new drops are added below.
4. Repeat 3 until all the new drops are added.

The new .gco file should have the same format and headers that those of the input file.

Chapter 9

Distinctive Features

The technology is relatively new. Therefore this is an explorative work trying to solve the hole filling problem, rather than improving a previous approach. Most of the comparison will be done with classic infill strategies, as presented in section 2.3. The main features are:

1. The filling is individualized to each area/slice. Unlike before as the idea was to change the whole filling without considering distinct areas. In this new implementation each void is treated individually.

2. The process works as an addition to previously known infill-strategies. Any infill can be processed.

3. The user can choose the degree of overlap for the void filling.

Also, the use of Voronoi diagrams is quite distinctive in the field of work. The slice filling phase in AM can be observed as a computational geometry and mathematical optimization problem, thus the use of tools and implements as the one proposed, can help in the improvement of the solutions.

Part III

Experiments

Introduction

In this part of the thesis two different experiments will be presented. The task was to develop a software tool that allow to fill, in different grades, the voids that appear in RP parts after the slice filling phase of the process. These experiments will test the efficacy and efficiency of the developed algorithm.

The procedure of the algorithm is to detect possible voids and according to a threshold fill them. This threshold limits the amount of overlapping that can be produced. This overlapping is the volume of new drops that fall into already deposited drops. A smaller the threshold will produce less overlapping but the voids will be less filled. The threshold is an input value that can be chosen by the user.

These experiments will compare parts with the new algorithm applied with parts that have not been processed by the new algorithm. Therefore the differences between the two kinds of parts will explain the effect of using this new implementation.

Two experiments will be performed. The first one will be a software test based on a full overlapping check of all the new drops added to fill the voids. This will allow to understand the effect of the overlapping threshold and obtain results about the optimal value to be chosen. Also, using the simulation capabilities of the algorithm the results of the algorithm can be seen before printing.

The second experiment will be a comparison of a set printed parts with algorithm and a set of printed parts without algorithm. Both sets will be composed of a number of parts of the same model. After printing, the parts will be viewed under the microscope with measuring capabilities to observe the differences on void size between parts with algorithm and parts without algorithm.

Chapter 10

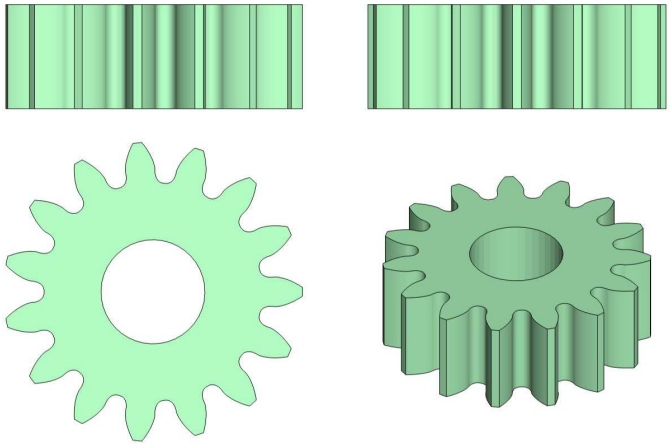
Software Experiment

To evaluate the functioning of the new implementation a software test is executed. Using diverse input files and configurations, the results will be evaluated according to the expectations.

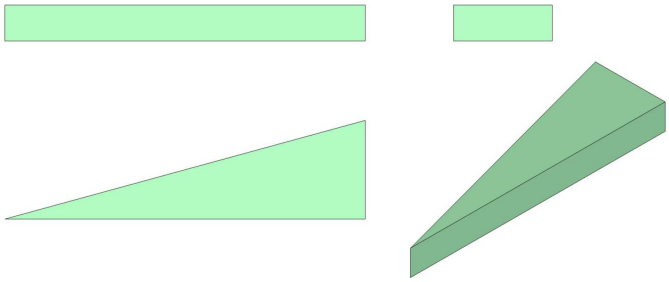
This files will be different models where different kinds of voids are produced. For each model the 3 available fillings (contours, hatchings and dense) will be tested, with and without algorithm. Consequently the difference of applying the algorithm or not will be observed.

The input value of the overlapping threshold will be also tested. This is the only configuration parameter that the user can choose that affect the result of the process. Therefore the effect of this parameter has to be tested in order to obtain an optimal value that produces optimal results.

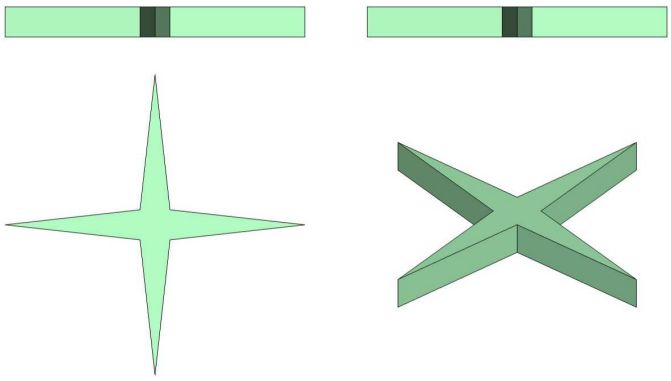
The models chosen for the test are presented in figure 10.1.



(a) Planet



(b) Spitz15



(c) Stern2

Figure 10.1: Models used in the test. The disposition of the pictures in each model is, from left to right and up to down: front, side, top and isometric.

10.1 Question

The goal of the test is to answer the following questions:

1. Is the implementation functioning correctly?
2. Which value for the overlapping threshold is optimal?

10.2 Hypothesis

The overlapping threshold is not to be surpassed. The condition for void filling prevents the deposition of drops that surpass the threshold. A surpassing of the threshold will indicate a major failure in the inner programming.

The effect of a higher overlapping threshold will be a higher number of new drops added. Resulting on a more complete filling of the voids at the cost of more overlapping.

10.3 Equipment

The test is implemented in MATLAB (2013b) and executed in a computer with the following specifications: processor i7-2630QM 2.00GHz and 8 gigabytes of RAM.

10.4 Idea

The overlapping between drops can be calculated with the linear approximation proposed in the previous chapter. In the previous chapter the overlapping between drops was calculated with a linear approximation. For this test an exact approach [47] for this calculation will be employed.

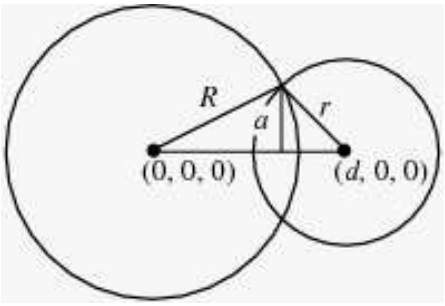


Figure 10.2: Intersection concept in 2D, [47]

The volume of a spherical cap that represents the three dimensional intersection of two spheres of the same radius "r" ($R = r$ in figure 10.2), separated a distance "d" is:

$$V = \frac{1}{12} \cdot \pi \cdot (4 \cdot r + d) \cdot (2 \cdot r - d)^2 \tag{10.1}$$

The test will measure the overlapping produced on the addition of new drops. This overlapping can be between initial drops and new drops or between new drops. The overlapping result is given in volume units consequently with the units of length used for the radius and distance.

The ratio of overlapped volume added in the implementation can be calculated dividing by the total volume added. The volume added is calculated through the volume of the sphere equation:

$$V = \frac{4}{3} \cdot \pi \cdot r^3 \tag{10.2}$$

The algorithm add new drops to the part being treated. This new drops can be counted and the total number of drops added is known. The product of the number of drops added with the volume of the sphere results in the total volume added.

The ratio of overlapped volume is called the overlapping of the object. This value should not surpass the overlapping threshold selected in the configuration.

Also for each execution of the implementation the result can be simulated to observe it before printing. This allows to decide if the overlapping is acceptable or not. This decision is made from the experience.

10.5 Process

This test is implemented outside the main program. It takes the initial drops and the new drops as input. It will give the overlapping ratio as output and also all the intermediate data.

As stated before, the test will measure the overlapping between new and initial drops and also between new drops, but will not measure the overlapping that may be already between initial drops. The calculation process is represented in the figure 10.3.

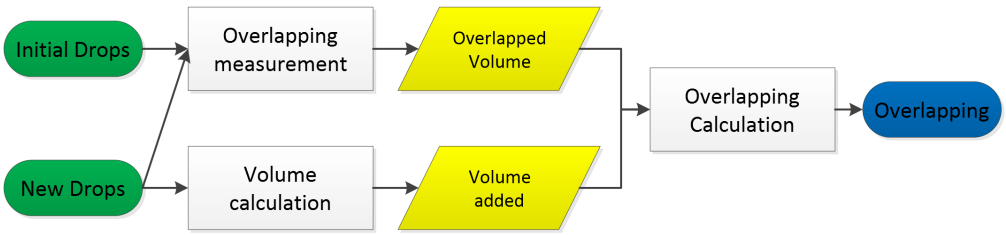


Figure 10.3: Process workflow

At the end of the test, all the resulting data is available for analysis.

10.6 Evaluation Methods

Once the overlapping is calculated, it will be compared with the overlapping threshold used. If the threshold is bigger than the resulting overlapping, the answer to the first question is no and the program functioned correctly.

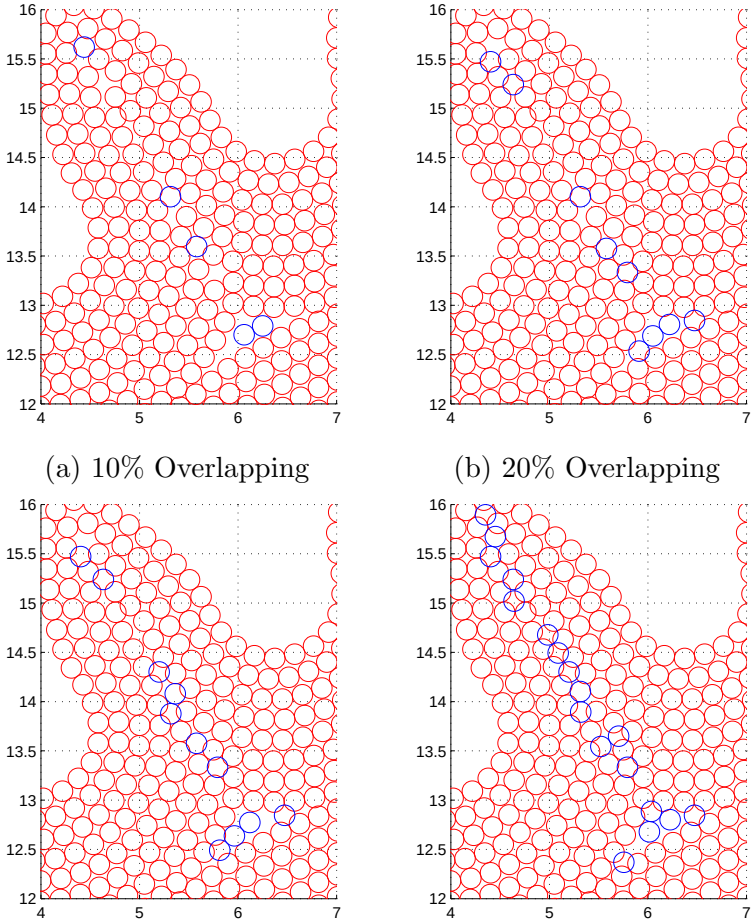
As the overlapping threshold increases, the increase of new volume added will be compared to the increase of overlapped volume.

10.7 Experiment Parameters

For this experiment, 3 different models were used, presented at the beginning of the chapter (Planet, Spitz15 and Stern2) in figure 10.1. A range of overlapping threshold was configured from 10% to 50% in steps of 10%. Each model was tested in all 3 filling methods available (hatching, contours and dense).

The reason behind this range of overlapping threshold is the experience of analyzing the simulations and observing the visual results. The ideal resulting overlapping will not surpass 10%. The overlapping results are not expected to surpass the threshold. Therefore a range composed of thresholds bigger than 10% is necessary. Also to observe the effect of too big thresholds.

To exemplify the visual effect of the variation of the overlapping, some simulations were executed (Figure 10.4).



(a) 10% Overlapping (b) 20% Overlapping

(c) 30% Overlapping (d) 40% Overlapping

Figure 10.4: Simulations with different overlapping thresholds, simulated in MATLAB

10.8 Experiment Results

The results are presented in the tables 10.1, 10.2 and 10.3.

The resulting overlapping value (6th column) never surpasses the chosen overlapping threshold (3rd column), in any of the filling cases and models. This confirms the properly functioning of the implementation.

The new volume added increases proportionally to the overlapping threshold. This is logical, because if a bigger threshold is chosen, the filling algorithm will have more space to build a filling solution and will be able to fit more drops. This will produce also an increase on the overlapped volume, as can be observed in the

Table 10.1: Contour filling results

Contours	**Initial Volume** $[mm^3]$	**Overlapping Threshold** [%]	**New Volume Added** $[mm^3]$	**Overlapped Volume Added** $[mm^3]$	**Overlapping** [%]
Planet	465,08	10	11,24	0,18	1,58
		20	19,16	1,17	6,11
		30	27,03	3,34	12,36
		40	38,09	7,70	20,22
		50	227,70	99,04	43,49
Spitz15	922,86	10	3,83	0,03	0,73
		20	9,72	0,56	5,76
		30	15,01	1,98	13,22
		40	18,27	3,38	18,51
		50	292,02	133,31	45,65
Stern2	310,21	10	6,20	0,09	1,44
		20	10,27	0,58	5,66
		30	12,60	1,16	9,18
		40	16,38	2,56	15,60
		50	133,46	59,46	44,56

Table 10.2: Hatching filling results

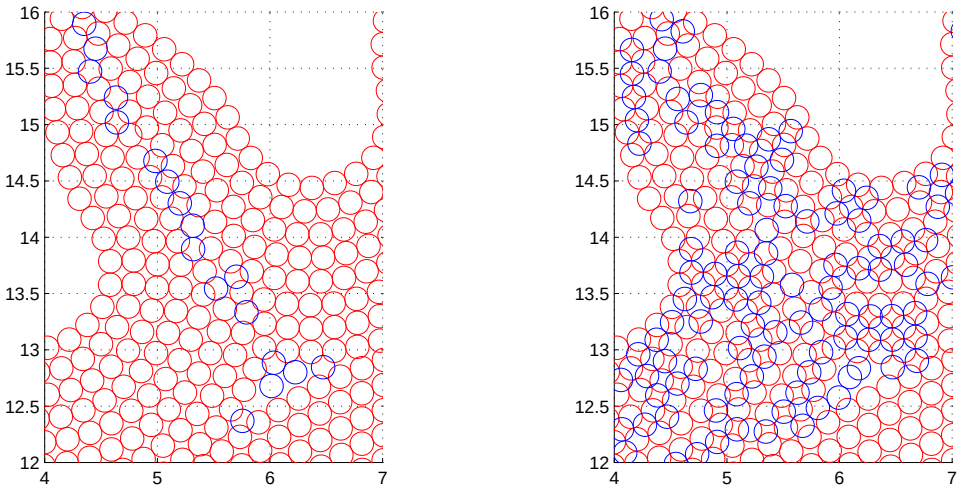
Hatching	**Initial Volume** $[mm^3]$	**Overlapping Threshold** [%]	**New Volume Added** $[mm^3]$	**Overlapped Volume Added** $[mm^3]$	**Overlapping** [%]
Planet	498,17	10	-	-	-
		20	0,11	0,02	18,70
		30	2,24	0,57	25,36
		40	7,10	2,39	33,73
		50	157,65	75,09	47,63
Spitz15	938,85	10	0,26	0,01	2,16
		20	2,24	0,31	13,86
		30	7,66	1,62	21,18
		40	12,34	3,53	28,63
		50	395,83	185,66	46,90
Stern2	322,34	10	0,61	0,01	1,22
		20	2,43	0,26	10,72
		30	6,57	1,29	19,68
		40	10,48	2,87	27,43
		50	119,55	55,46	46,39

Table 10.3: Dense filling results

Dense	**Initial Volume** $[mm^3]$	**Overlapping Threshold** [%]	**New Volume Added** $[mm^3]$	**Overlapped Volume Added** $[mm^3]$	**Overlapping** [%]
Planet	498,17	10	0,07	0,00	1,44
		20	7,55	1,01	13,33
		30	18,07	3,85	21,32
		40	27,69	7,62	27,52
		50	70,42	28,37	40,29
Spitz15	929,08	10	0,30	0,01	2,26
		20	3,72	0,47	12,60
		30	11,37	2,57	22,59
		40	21,16	6,12	28,92
		50	48,44	18,74	38,68
Stern2	315,76	10	0,66	0,01	1,37
		20	4,01	0,44	10,94
		30	10,30	2,03	19,70
		40	16,15	4,20	25,99
		50	45,78	18,84	41,15

table.

A distinctive part of the data is the results for the highest threshold tested. The 50% test give a big increase in new volume added when compared with the previous step of 40%. Not only the new added volume increases considerably but also the overlapped volume gets a big increase. The overlapping results give what can be seen as too high values of overlapping. Almost half of the volume being added is overlapped volume. A first impression would be that this is not a very efficient solution for filling voids. To check this situation, simulations (Figure 10.5) for the 40% and 50% overlapping threshold will be executed.



(a) Planet part with 40% overlapping threshold

(b) Planet part with 50% overlapping threshold

Figure 10.5: Critical situtation in void filling

It can be observed that in the 50% case smaller voids are being filled. The problem is that those small voids are not to be filled. Those voids are the space between drops that cannot be put nearer. This means that this space is supposed to be empty because that is the densest packing of drops that the filling method for this machine can achieve. Trying to fill this voids will produce wrong parts.

As stated before, the higher the threshold the smaller are the voids that can be filled, but as the simulations show, filling too small voids is not a good result.

Another result to be commented is that each model and filling produces different values. The reason behind is that each model has a different geometry and therefore will produce different kinds of voids that will be filled differently.

10.9 Experiment Conclusions

The implementation developed in this thesis works accordingly to the expectations. No failure was found as the overlapping threshold was never surpassed.

Allowing more overlap brings an increase of the volume filled while more of the volume added is overlapped volume. The size of the voids filled depends on the overlapping threshold allowed. A higher threshold will allow smaller voids to be filled.

Observing only the data present in the tables is not possible to make a statement about the value of the optimal overlapping threshold. Obtaining the data an overlapping of 10% is recommended, but it would be also necessary that the user runs simulations to decide which overlapping threshold fits his needs inside the previously proposed range of 10%-40% of overlapping threshold.

Chapter 11

Physical Experiment

In this experiment, parts printed with the new implementation will be viewed under the microscope. The chosen model to be printed will be "Planet" (Figure 10.1a and 11.1), the reason is that its geometry produces various kinds of voids, as outer-inner void[1], shape change[2], and acute corner[3].

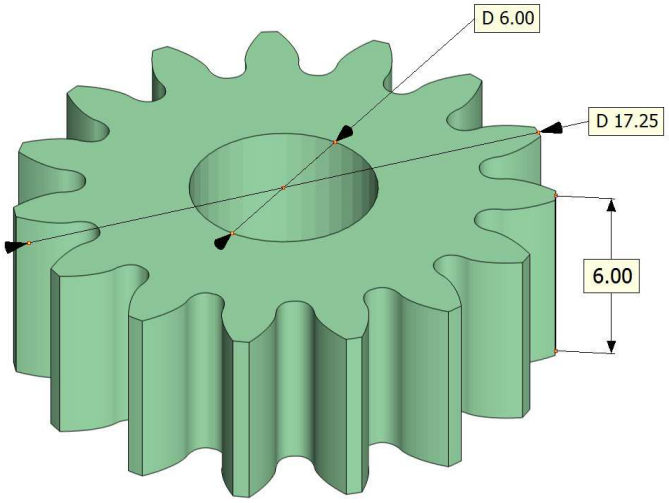


Figure 11.1: Main measures of the "Planet" model, values in mm

Four sets of this model will be printed. Two sets with contour filling method, one without the algorithm and the other with the algorithm applied. The other two sets will be printed with the dense filling method, again one without algorithm and the other with algorithm. The overlapping threshold chosen for the sample

[1]Page 34
[2]Page 35
[3]Page 36

was 20%. It was estimated to be adequate since it produces an overlapping near but below the recommended 10%, as seen in tables 10.1 and 10.3. The parts will be printed with a droplet deposition machine, like the one previously presented[4]. The material used to print the parts is BASF TERLURAN GP-35. This experiment will allow to verify the real effectivity of the new implementation.

11.1 Question

The question for this experiment will be: are there visible improvements in the parts printed with the new algorithm?

11.2 Hypothesis

As seen in the simulations the new drops are going to be deposited inside the found voids. Therefore, there will be a decrease of the size of the voids seen under the microscope.

11.3 Equipment and Sample

Two different microscopes were used during the experiment:

- Conrad DP-M17 USB Digital Microscope.
- Wild Heerbrugg M5A

The Conrad DP-M17 was used to execute an initial visual analysis of the printed parts to select those with the highest number of voids to be viewed afterwards under the other microscope. It was also used to take pictures to visually compare the parts.

The M5A was used to measure the size of the voids appearing in the printed parts with and without the new implementation.

The camera attached to the M5A used to take the pictures is a Moticam 1000 1.3M Pixel USB2.0. The software used to process the pictures is the Motic Images Plus 2.0 ML.

[4]Page 14

11.4 Measurement Parameters

The parameter expected to change is the size of the voids that appear in the parts. To measure that change, the surface of the voids in the top most layer will be quantified with the help of the M5A microscope.

11.5 Evaluation Methods

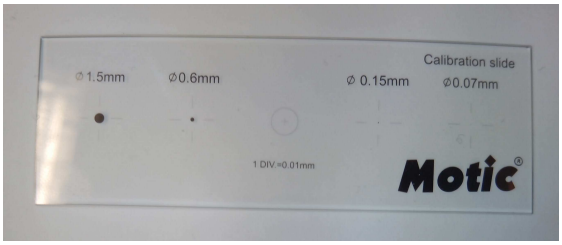
The evaluation under the M5A of every printed part for the experiment is a hard and time consuming task. A high quantity of the voids appearing in the parts are really small and consequently the measuring process, presented in the next section, would take too much time and wouldn't be precise enough if all the parts were to be fully analyzed. To obtain more precise results on the measuring, only the biggest voids will be measured, as they are the easier to see.

Therefore, 3 parts from each set were selected. The criterion for this selection is for the parts to have the highest quantity of visually noticeable voids. These selected parts were viewed under the M5A microscope.

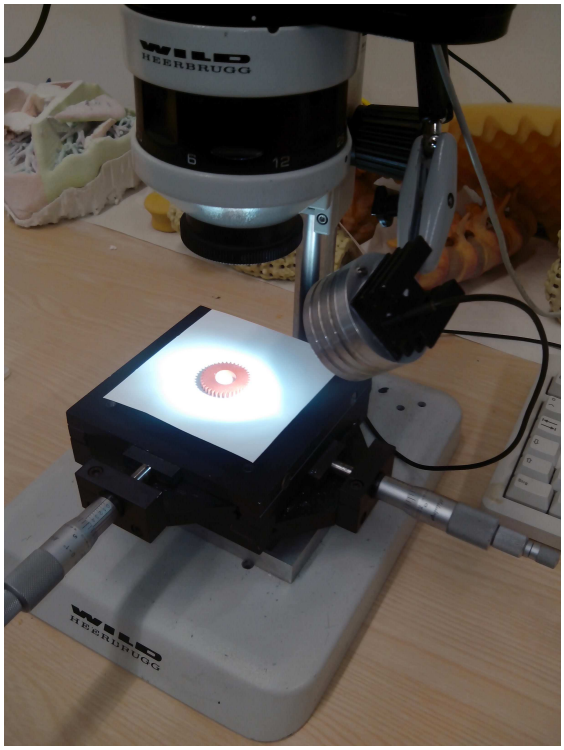
The biggest surfaces were measured. From those values the highest of each set is selected for comparison between sets of the same filling method.

11.6 Process

The measuring process with the M5A microscope stands as follows:



(a) Calibration plate



(b) Microscope Wild Heerbrugg M5A

Figure 11.2: Part of the equipment used in the experiment

1. Calibration of the microscope: to obtain correct measurements, the microscope must be calibrated. With the help of a calibration plate (Figure 11.2a) the microscope is calibrated for an amplification value (in this case 25x) and a focus configuration. If the amplification or the focus change, the calibration must be redone.

2. Inspection: the part is set in the plate of the microscope (Figure 11.2b) and the objective is fixed over the void.

3. Measuring: a photograph of the void is taken. The software of the camera measures the void surface (Figure 11.3).

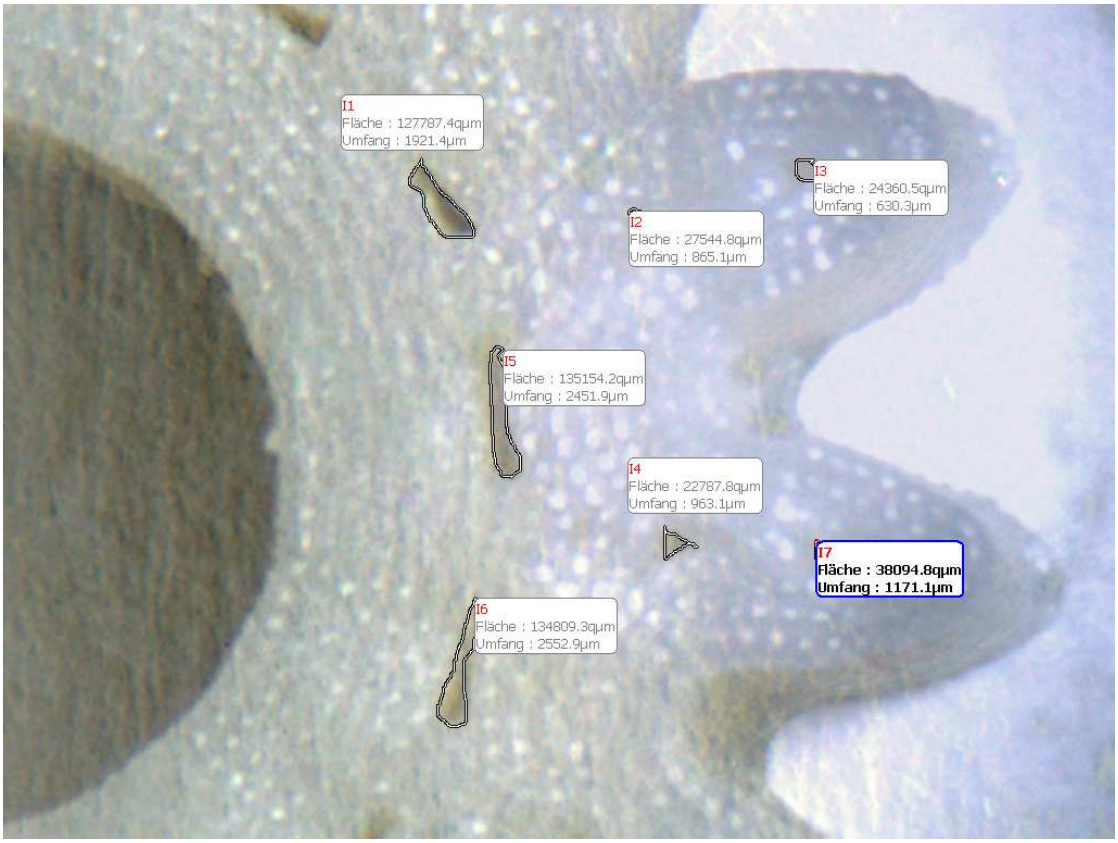
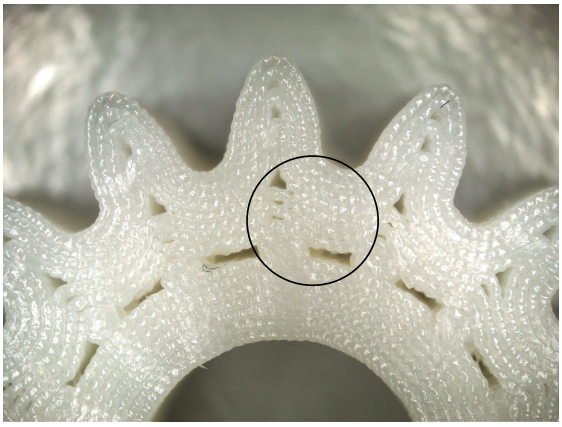


Figure 11.3: Measuring of voids

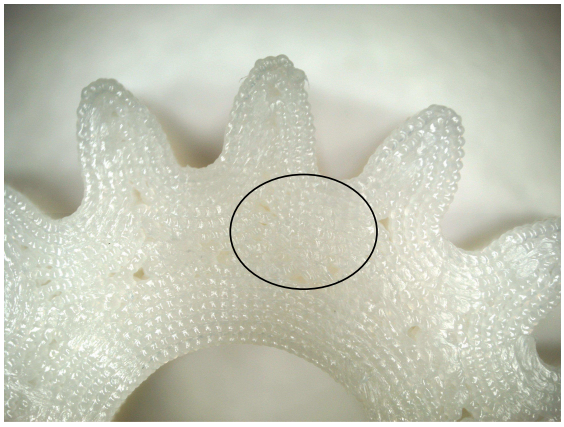
For each measure the data is stored in a spreadsheet for later analysis.

11.7 Experiment Results

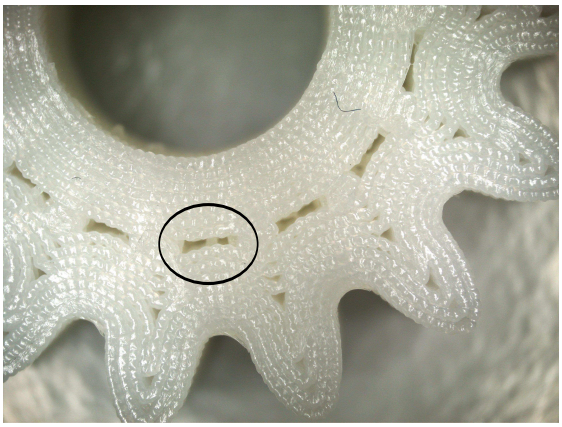
In the figures 11.4 and 11.5 can be observed the difference between parts with algorithm and parts without.



(a) Planet in contour filling without algorithm, upper part



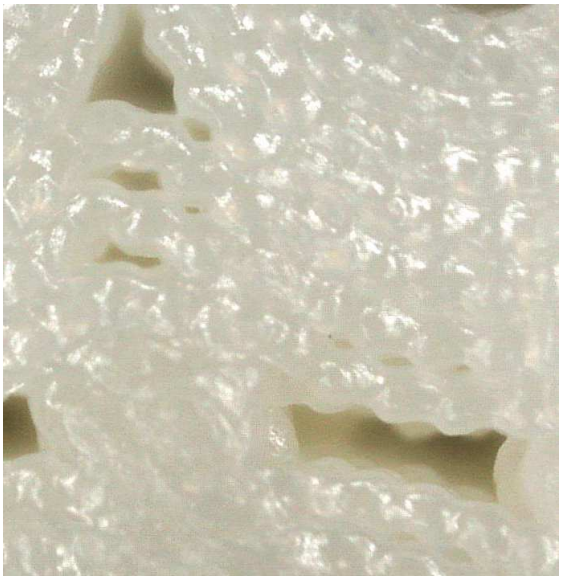
(b) Planet in contour filling with algorithm, upper part



(c) Planet in contour filling without algorithm, lower part

(d) Planet in contour filling with algorithm, lower part

Figure 11.4: Parts without algorithm (left) compared to parts with algorithm (right), the areas marked in black are zoomed below.



(a) Planet in contour filling without algorithm, upper part, zoomed area

(b) Planet in contour filling with algorithm, upper part, zoomed area

(c) Planet in contour filling without algorithm, lower part, zoomed area

(d) Planet in contour filling with algorithm, lower part, zoomed area

Figure 11.5: Parts without algorithm (left) compared to parts with algorithm (right), , zoomed areas.

The experiment gives the following results (Table 11.1 and figure 11.6):

In both cases the void maximal surface has decreased. The differences are, for dense filling 53.17% and for contour filling 71.63%.

Table 11.1: Functional Experiment Results

	Max. Surface [μm^2]	
	Without Algorithm	**With Algorithm**
Dense	285085,3	133491,5
Contours	305076,2	86548,8

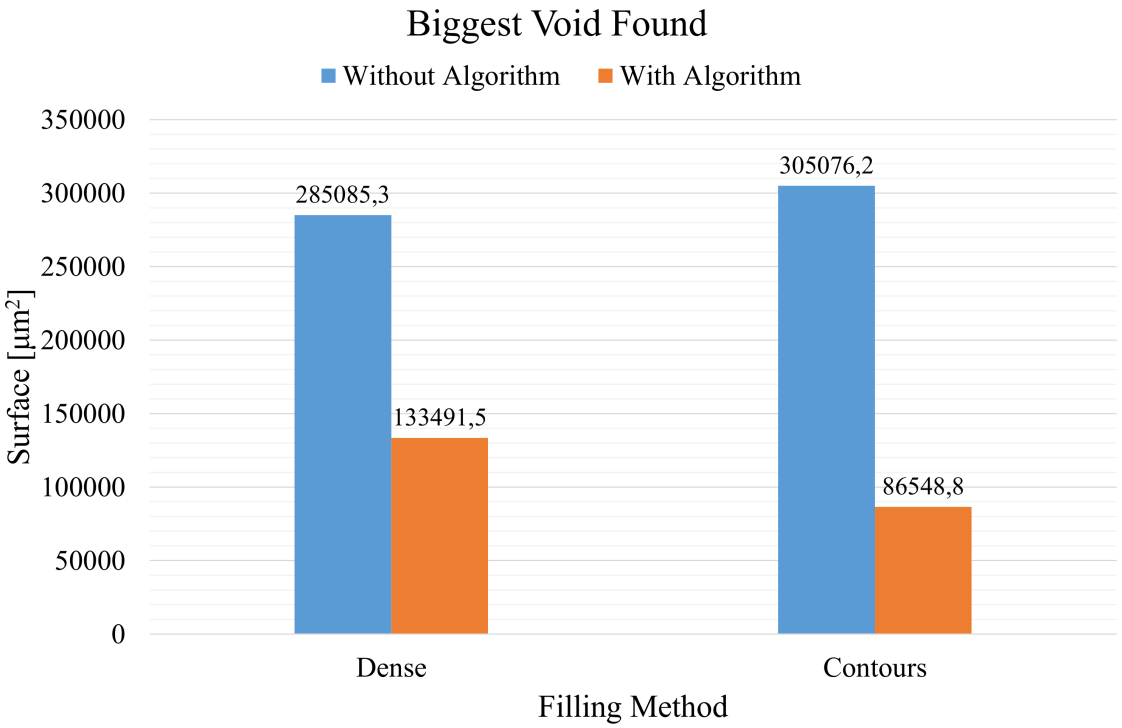


Figure 11.6: Graph of results

11.8 Experiment Conclusions

The experiment give the expected results. It decreases effectively the size of the voids, satisfying the task for which it was implemented.

Still there are voids as the images (Figure 11.4) show. That is the effect of the overlapping threshold and a higher threshold will fill those remaining voids.

Nevertheless there is a visible improvement, the parts with the algorithm present smaller voids and we can conclude that the new implementation is effective.

Part IV

Summary

Rapid prototyping (RP) technology offers engineers and designers the capacity of producing models and prototypes in a quick manner and directly from the CAD model. As presented in the state of the art chapter, currently on the market there is a diverse variety of technologies, each one with different advantages and disadvantages, from which the engineer can choose according to the requirements of the product. The benefits of this technology can be applied in diverse industries, as automotive, aeronautical and medical industry.

The RP process has various phases that can be divided in a software stage and a fabrication stage. In turn the software stage has also phases and one of them is the main core of this thesis, the slice filling phase. This phase is the last one before going on with the fabrication stage. The slice filling phase will produce a file in .gco format that the machine will be able to read to produce the physical object.

In this thesis the focus has been the droplet deposition technology. The physical characteristics of droplets require another different approach, than those that have been used for the more common filament deposition technologies. This technology requires also a different software in the processing of the CAD model.

The research done about the state of the art in this field includes a study about the different filling methods available. This filling methods differ on how the slices are filled with deposited material.

Another part of the research was the packing problem approach. This mathematical problem adapts to the goal of the task, as the slice have to be filled the better way possible. The analysis of the current approaches to solve this problem turn out that the computation time required is too high to be considered a feasible solution for RP.

The appearance of voids is one of the limitations of this technology. Different filling methods produce different kinds of voids. This was the matter of study in the limitations of the state of the art research. It covered the filling methods available, searching for the kinds of voids that they produce. The droplet fillings were as well studied with the help of simulations.

The task in this thesis is to develop a software tool that overcomes that limitation, filling the voids that are produced after the slices are filled by previous software. This new implementation is divided in four modules: parser, analyzer, hole filling algorithm and Gcode writer.

The input of the program is the processed .gco file, obtained from a previous software that carry out the slice filling phase. This file is going to be read by the parser. Then the information obtained will be analyzed and prepared by the analyzer. The hole filling algorithm will then detect the possible holes/voids and fill them according to a threshold. Finally the new drops will be added to a new .gco file along with the initial drops. This threshold is the configuration parameter

of the program. It allows the user to choose how extensively the voids are filled.

For the hole filling algorithm two approaches were tested: a nearest neighbor approach and a Voronoi diagram approach. The comparison gave that the Voronoi approach was faster. Therefore it was chosen for the final algorithm.

Another capability of this program is the simulation of the droplet-formed model. This allows the user to see how the machine will lay the drops in each slice.

Once the algorithm was finished and to test its functioning, two experiments were executed. A software test checks the correct function of the program and also intends to find an optimal value for the overlapping threshold. With different models, fillings and thresholds the results were analyzed. An overlapping calculation is developed to measure how much volume added is being overlapped. This analysis results in a positive functioning but not in a definitive value of the overlapping threshold. The 10% value is proposed as a tip for the user. To choose a final value for the threshold the simulation tool must be used to check the expected result and then decide the value.

The second experiment involves printing a certain model in the machine to physically observe the results of the algorithm. Therefore, four set of parts were printed in two different filling methods, two without algorithm and two with algorithm. With the help of a microscope the size of the voids were measured and the variation effect of the algorithm analyzed.

The results were a decrease on the size of the voids when comparing the parts without algorithm and the parts with algorithm. This represents a positive fulfillment of the proposed task. Pictures from the resulting parts were taken.

Currently the program is stand-alone and implemented in a MATLAB environment. Since the task is accomplished, as future improvement, the implementation in C++ of the program would allow it to be included in a bigger software, like the one that was used to produce the input files. Some of the functions used (Voronoi diagrams and geometry functions) are part of the MATLAB environment and a substitute should be found for them.

Bibliography

[1] Grimm, T. (2004): "User's Guide to Rapid Prototyping", *Society of Manufacturing Engineers*.

[2] Junk, S.; Sämann-Sun, J.; Niederhofer, M. (2010): "Application of 3D Printing for the Rapid Tooling of Thermoforming Moulds", *Springer London*.

[3] Moon, S. K.; Tan, Y. E.; Hwang, J.; Yoon, Y. (2014): "Application of 3D printing technology for designing light-weight unmanned aerial vehicle wing structures", *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol.1, num.3, pp. 223-228.

[4] Stratasys (2013):"Additive Manufacturing Reduces Tooling Cost and Lead Time to Produce Composite Aerospace Parts", `http://www.stratasys.com/~/media/Case%20Studies/Aerospace/SSYS-CS_Fortus-ACS-05-13.pdf`.

[5] Straub, J. (2014): "Design and fabrication of reconstructive mandibular models using fused deposition modeling", *Aerospace Conference, 2014 IEEE*, pp. 1–7.

[6] Bibb, R.; Eggbeer, D.; Evans, P. (2009): "Rapid prototyping technologies in soft tissue facial prosthetics: current state of the art", *Rapid Prototyping Journal*, vol.16, num.2, pp. 130–137.

[7] Chimento, J.; Highsmith, M. J.; Crane, N. (2011): "3D printed tooling for thermoforming of medical devices", *Rapid Prototyping Journal*, vol.17, num.5, pp. 387–392.

[8] Kouhi, E.; Masood, S.; Morsi, Y. (2008): "Design and fabrication of reconstructive mandibular models using fused deposition modeling", *Assembly Automation*, vol.28, num.3, pp. 246–254.

[9] Pullen, D.; Sapuan, S. M. (2005): "Concurrent design and manufacturing process of automotive composite components", *Assembly Automation*, vol.25, num.2, pp. 146-152.

[10] Gibson, I.; Rosen, D. W.; Stucker, B. (2010): "Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing", *Springer*.

[11] Materialgeeza (2008):"Stereolithography apparatus", Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons., `https://commons.wikimedia.org/wiki/File:Stereolithography_apparatus.jpg`.

[12] Materialgeeza (2008):"Selective laser melting system schematic", Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons., `https://commons.wikimedia.org/wiki/File:Selective_laser_melting_system_schematic.jpg`.

[13] Kaidor (2013):"Hole Force Lines", Own work based on File:HoleForceLines.gif. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons., `https://commons.wikimedia.org/wiki/File:HoleForceLines.svg`.

[14] MakerBot (2014):"MAKERBOT REPLICATOR MINI", MakerBot® Industries, LLC, `http://store.makerbot.com/replicator-mini`.

[15] Chua, C. K.; Leong, K. F.; Lim, C. S. (2010): "Rapid Prototyping: Principles and Applications", *Society of Manufacturing Engineers*.

[16] Liou, F. W. (2008): "Rapid prototyping and engineering applications: a toolbox for prototype development", *CRC Press*.

[17] Szabó, P. G.; Markót, M. Cs.; Csendes, T.; Specht, E.; Casado, L. G.; García, I. (2007): "New Approaches to Circle Packing in a Square", *Springer*.

[18] Turner, B. T.; Strong, R.; Gold, S. A. (2014): "A review of melt extrusion additive manufacturing processes: I. Process design and modeling", *Rapid Prototyping Journal*, vol.20, num.3, pp. 192–204.

[19] Schwaiger, J; Lüth, T. C.; Irlinger, F. (2013): "G-Code Generation for a New Printing Process Based on 3D Plastic Polymer Droplet Generation", *Proceedings of the ASME 2013 International Mechanical Engineering Congress & Exposition*, .

[20] Schwaiger, J. (2013): "Codegenerierung für einen neuen 3D-Druckprozess auf Tropfenbasis", *TUM MiMed*, Doctoral thesis to be published.

[21] Jin, G. Q.; Li, W. D.; Tsai, C. F.; Wang, L. (2011): "Adaptive tool-path generation of rapid prototyping for complex product models", *Journal of Manufacturing Systems*, vol.30, pp. 154–164.

[22] Jin, G. Q.; Li, W. D.; Gao, L. (2013): "An adaptive process planning approach of rapid prototyping and manufacturing", *Robotics and Computer Integrated Manufacturing*, vol.29, pp. 23–38.

[23] Kulkarni, P.; Marsan, A.; Dutta, D. (2000): "A review of process planning techniques in layered manufacturing", *Rapid Prototyping Journal*, vol.6, num.1, pp. 18–35.

[24] Kulkarni, P.; Dutta, D. (1999): "Deposition Strategies and Resulting Part Stiffnesses in Fused Deposition Modeling", *Journal of Manufacturing Science and Engineering*, vol.121, pp. 93–103.

[25] Agarwala, M. K.; Jamalabad, V. R.; Langrana, N. A.; Safari, A.; Whalen, P. J.; Danforth, S. C. (1996): "Structural quality of parts processed by fused deposition", *Rapid Prototyping Journal*, vol.2, num.4, pp. 4–19.

[26] Wenbiao, H.; Jafari, M. A.; Danforth, S. C.; Safari, A. (2002): "Tool Path-Based Deposition Planning in Fused Deposition Processes", *Journal of Manufacturing Science and Engineering*, vol.124, pp. 462–472.

[27] Zhou, M. Y. (2004): "Path planning of functionally graded material objects for layered manufacturing", *International Journal of Production Research*, vol.42, num.2, pp. 405–415.

[28] Nancharaiah, T.; Raju, D. R.; Raju, V.R. (2010): "An experimental investigation on surface quality and dimensional accuracy of FDM components", *International Journal on Emerging Technologies*, vol.1, pp. 106–111.

[29] Nevin, H.; Mehrdad, H. (2014): "Deposition direction-dependent failure criteria for fused deposition modeling polycarbonate", *Rapid Prototyping Journal*, vol.20, num.3, pp. 221–227.

[30] Han, W.; Jafari, M. A.; Seyed, K. (2003): "Process speeding up via deposition planning in fused deposition-based layered manufacturing processes", *Rapid Prototyping Journal*, vol.9, num.4, pp. 212–218.

[31] Qiu, D.; Langrana, N. A.; Danforth, S. C.; Safari, A.; Jafari, M. (2001): "Intelligent Toolpath for Extrusion-based LM Process", *Rapid Prototyping Journal*, vol.7, num.1, pp. 18–23.

[32] Yang, Y.; Loh, H. T.; Fuh, J. Y. H.; Wang, Y. G. (2002): "Equidistant path generation for improving scanning efficiency in layered manufacturing", *Rapid Prototyping Journal*, vol.8, num.1, pp. 30–37.

[33] Bertoldi, M.; Yardimci, M. A.; Pistor, C. M.; Güçeri, S. I. (1998): "Domain Decomposition and Space Filling Curves in Toolpath Planning and Generation", *Solid Freeform Fabrication Symposium*, pp. 267–276.

[34] Prša, J.; Schwaiger, J.; Irlinger, F.; Lüth, T. C. (2013): "Dense 3D-Packing Algorithm for Filling the Offset Contours of a New Printing Process Based on 3D Plastic Droplet Generation", *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 74–78.

[35] van Weeren, R.; Agarwala, M. K.; Jamalabad, V. R.; Bandyophadyay, A.; Vaidyanathan, R.; Langrana, N. A.; Safari, A.; Whalen, P.; Danforth, S.; Ballard, C. (1995): "Quality of Parts Processed by Fused Deposition", *Solid Freeform Fabrication Symposium*, pp. 314–321.

[36] Volpato, N.; Silveira, J. H.; de Souza, T. R. (2005): "A Process Planning Applicative for Rapid Prototyping Technology", *18th International Congress of Mechanical Engineering*, .

[37] Birgin, E. G.; Gentil, J. M. (2010): "New and improved results for packing identical unitary radius circles within triangles, rectangles and strips", *Computers & Operations Research*, vol.37, pp. 1318–1327.

[38] Galiev, S. I.; Lisafina, M. S. (2013): "Linear models for the approximate solution of the problem of packing equal circles into a given domain", *European Journal of Operational Research*, vol.230, pp. 505–514.

[39] Hifi, M.; M'Hallah, R. (2009): "A Literature Review on Circle and Sphere Packing Problems: Models and Methodologies", *Advances in Operations Research*, .

[40] Brooks, H. L.; Rennie, A. E. W.; Abram, T. N.; McGovern, J.; Caron, F. (2011): "Variable Fused Deposition Modelling – Analysis of Benefits, Concept Design and Tool Path Generation", *Proceedings of the 5th International Conference on Advanced Research in Virtual and Rapid Prototyping*, pp. 511–517.

[41] Qiu, D.; Langrana, N. A. (2002): "Void Eliminating Toolpath for Extrusion-based Multi-material Layered Manufacturing", *Rapid Prototyping Journal*, vol.8, num.1, pp. 38–45.

[42] Gooch, J. W. (2007): "Encyclopedic Dictionary of Polymers", *Springer*.

[43] Bendels, G. H.; Schnabel, R.; Klein, R. (2006): "Detecting Holes in Point Set Surfaces", *Journal of WSCG*, vol.14, .

[44] Bi, K.; Gu, N.; Tu, K.; Dong, W. (2006): "Neighborhood-Based Distributed Topological Hole Detection Algorithm in Sensor Networks", *International Conference on Wireless, Mobile & Multimedia Networks*, .

[45] Zhang, Y.; Zhang, X.; Wang, Z.; Liu, H. (2013): "Virtual Edge Based Coverage Hole Detection Algorithm in Wireless Sensor Networks", *IEEE Wireless Communications and Networking Conference*, pp. 1488–1492.

[46] Weisstein, E. W.: "Voronoi Diagram", *From MathWorld–A Wolfram Web Resource*, `http://mathworld.wolfram.com/VoronoiDiagram.html`.

[47] Weisstein, E. W.: "Sphere", *From MathWorld–A Wolfram Web Resource*, `http://mathworld.wolfram.com/Sphere.html`.

List of Figures

List of Tables

Anexo B

Concepto de Solapamiento

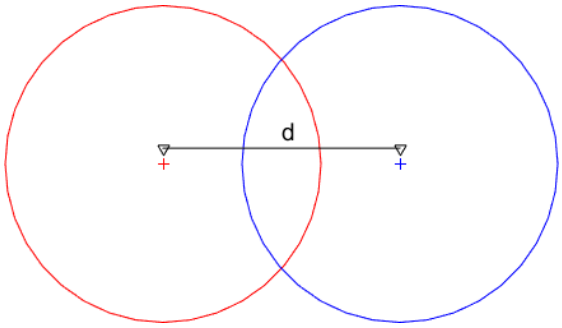


Figura B.1: Gotas situadas a una distancia d se solapan

El solapamiento es un efecto del rellenado efectivo de vacíos. Si no se permite, muchos vacíos podrían quedar sin rellenar. En el software previo desarrollado en el instituto el solapamiento se mide de manera lineal, como se presenta en la siguiente ecuación, siendo d la distancia entre los centros de dos gotas de radio r:

$$O(d = 0) = 100\,\% \tag{B.1}$$

$$O(d = 2 \cdot r) = 0\,\% \tag{B.2}$$

$$O[\,\%] = 100 \cdot \left(1 - \frac{d}{2 \cdot r}\right) \tag{B.3}$$

Así pues, si se limita el solapamiento a un cierto grado o umbral, se define una distancia mínima que las gotas tienen que respetar entre sí.

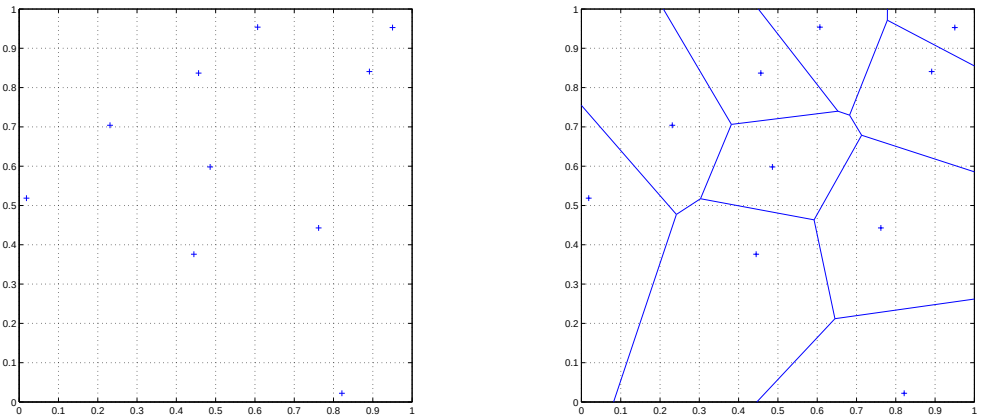
$$O < O_t \rightarrow d > d_t = 2 \cdot r \cdot \left(1 - \frac{O_t}{100}\right) \tag{B.4}$$

Anexo C

Diagramas de Voronoi

Como se define en [36], un diagrama de Voronoi es:

La partición de un plano con n puntos en polígonos convexos tal que cada polígono contiene exactamente un punto generador y todo punto en un polígono dado tiene al punto generador de dicho polígono como el más cercano que todos los otros puntos generadores. Un diagrama de Voronoi es llamado también como teselado de Dirichlet. Las celdas son llamadas regiones de Dirichlet, polígonos de Thiessen o polígonos de Voronoi.



(a) Nube de puntos generadores (b) Diagrama correspondiente a esos puntos

Figura C.1: Diagrama de Voronoi generado en MATLAB

En este trabajo los vértices de los polígonos se definen como vértices de Voronoi. Estos vértices pueden ser extraídos y aislados. La propiedad fundamental de estos vértices es que tienen la misma distancia a todos los puntos generadores de las celdas a las que pertenece.

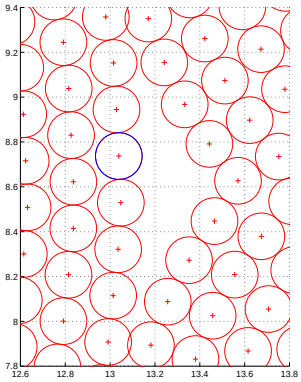
Anexo D

Ejemplo de Ejecución

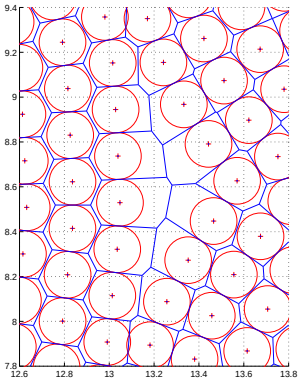
En este anexo se presenta una ejecución completa de ejemplo en la detección y rellenado de un vacío en una capa. En esta ejecución se emplea el método de los diagramas de Voronoi. En la primera sección se trata la detección del vacío, en la segunda el rellenado de este vacío. La razón de la elección de los diagramas de Voronoi sobre la aproximación por vecinos más cercanos se presenta en la tercera sección de este anexo, con una comparativa de tiempos de ejecución. En la cuarta sección se presenta un ejemplo de detección mediante vecinos más cercanos.

Para facilitar la comprensión de la impresión de gotas se presenta en la quinta sección 3 ejemplos, uno de cada tipo de rellenado. Por último el efecto del grado de solapamiento se presenta con un ejemplo en la última sección.

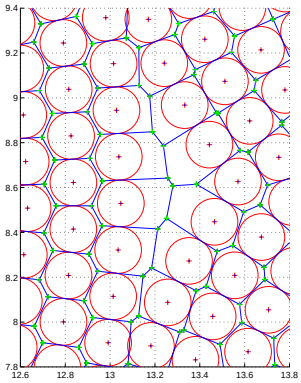
D.1. Generación del Diagrama de Voronoi



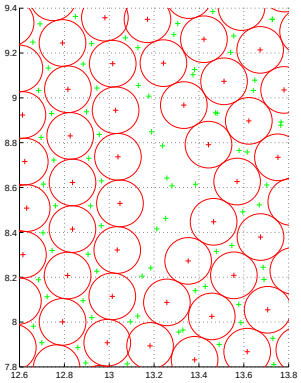
(a) Vacío a detectar



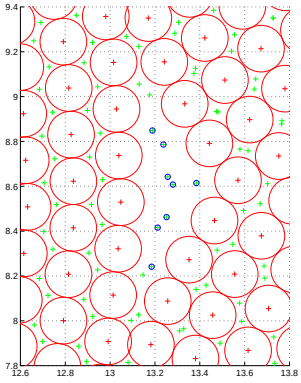
(b) Diagrama de Voronoi



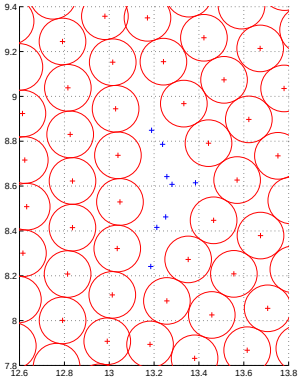
(c) Vértices de los polígonos



(d) Vértices de Voronoi



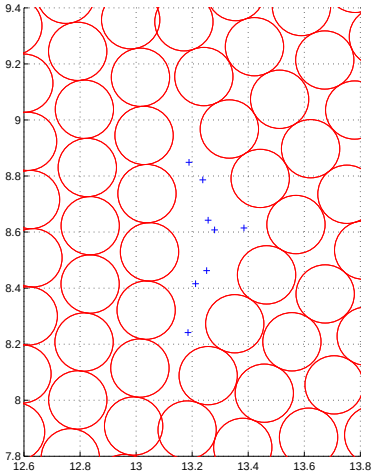
(e) Filtrado de los puntos de acuerdo al grado de solapamiento



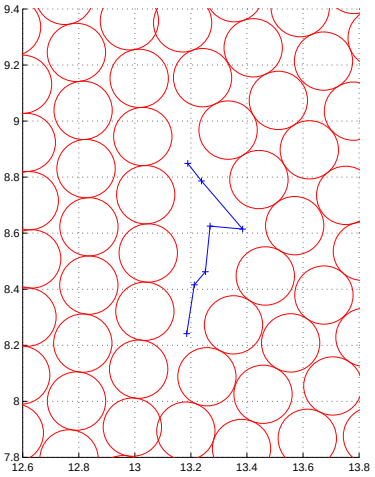
(f) Puntos resultantes

Figura D.1: Proceso del método con Voronoi, simulado en MATLAB

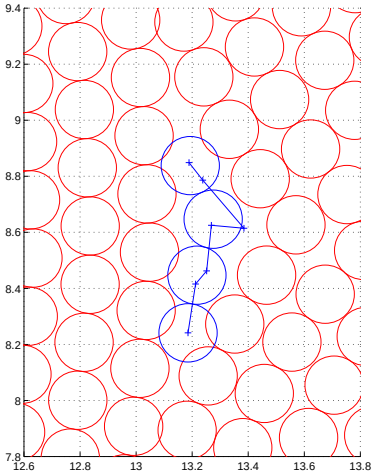
D.2. Rellenado del Vacío



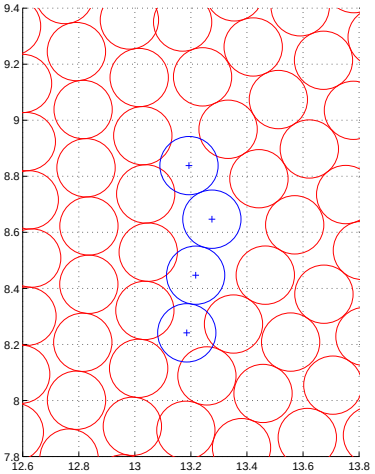
(a) Puntos en el vacío



(b) Ordenación y construcción de la linea a rellenar



(c) Rellenado de la linea



(d) Resultado de rellenado

Figura D.2: Proceso de rellenado, simulado en MATLAB

D.3. Comparativa de Aproximaciones

Para comparar los dos métodos propuestos para la detección de vacíos se mide el tiempo de ejecución en el procesado de una capa de 3 modelos diferentes, con los 3 rellenados diferentes en cada caso.

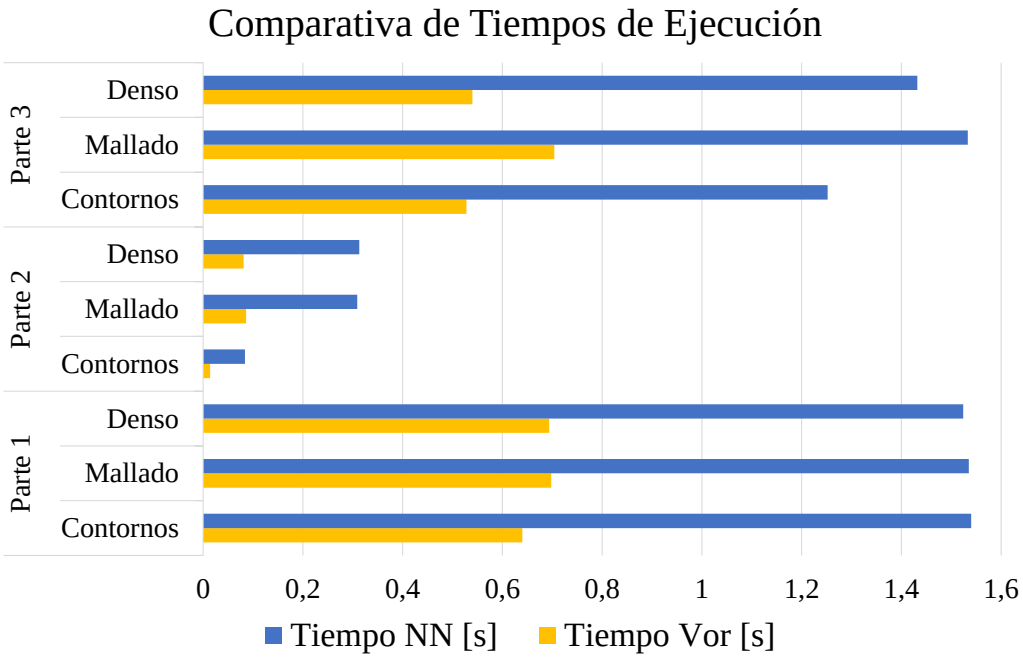
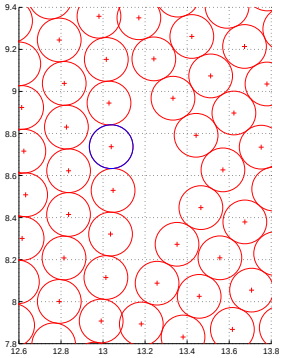


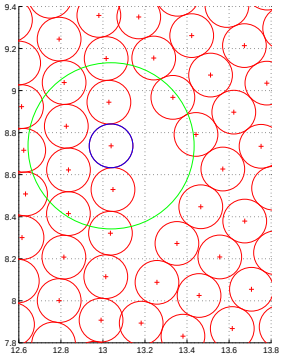
Figura D.3: Resultados de la Comparativa

Se observa en la gráfica que en todos los casos el uso de Diagramas de Voronoi (Vor) es más rápido que el método de Vecinos más Cercanos (NN).

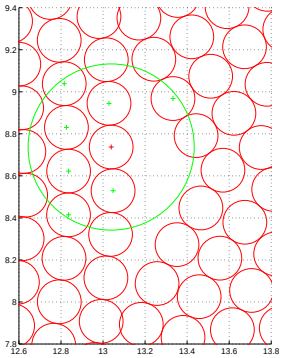
D.4. Vecinos más cercanos



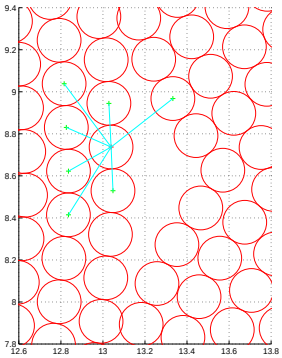
(a) Vacío a detectar



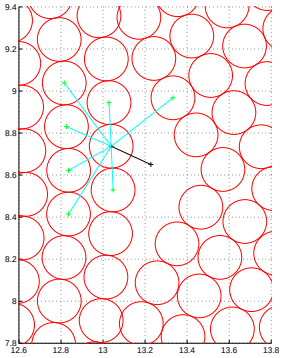
(b) Área de posibles vecinos



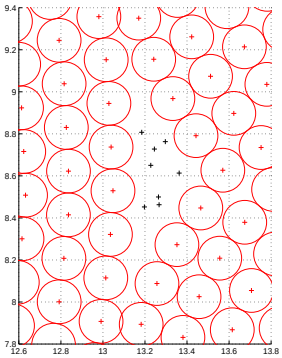
(c) Vecinos más cercanos detectados



(d) Vectores a los vecinos mas cercanos



(e) Puntos calculados en los bisectores de los ángulos más grandes



(f) Puntos resultantes

Figura D.4: Proceso de vecinos más cercanos, simulado en MATLAB

D.5. Ejemplos de rellenado

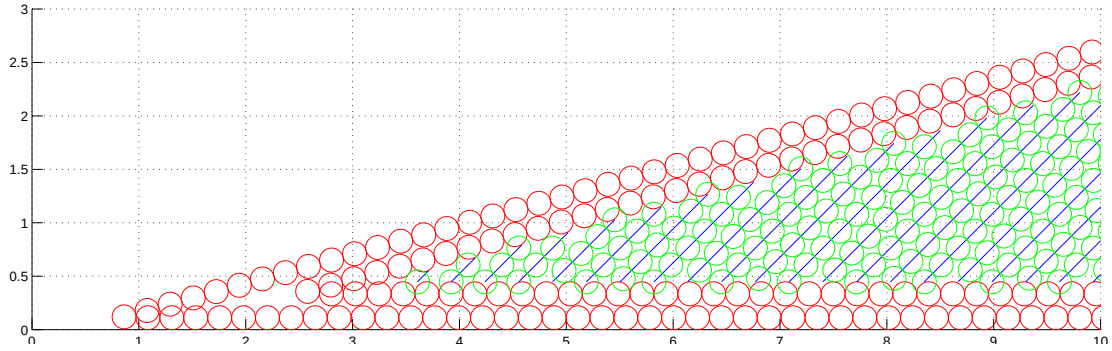


Figura D.5: Ejemplo de rellenado por mallado, simulado con MATLAB

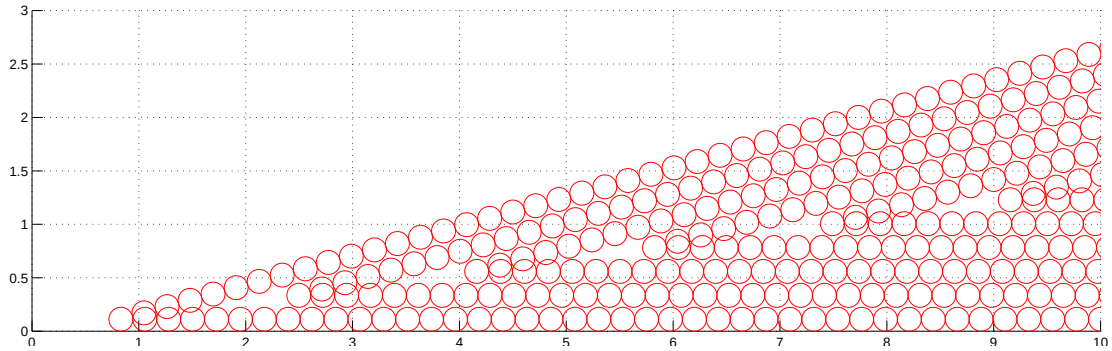


Figura D.6: Ejemplo de rellenado por contornos, simulado con MATLAB

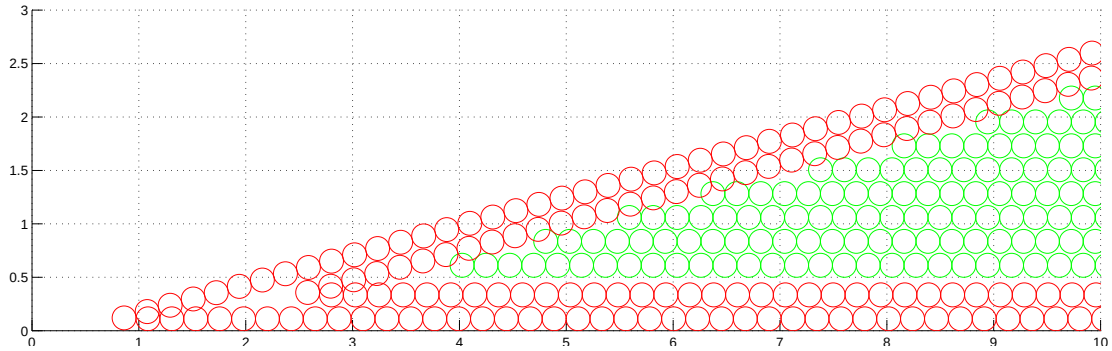
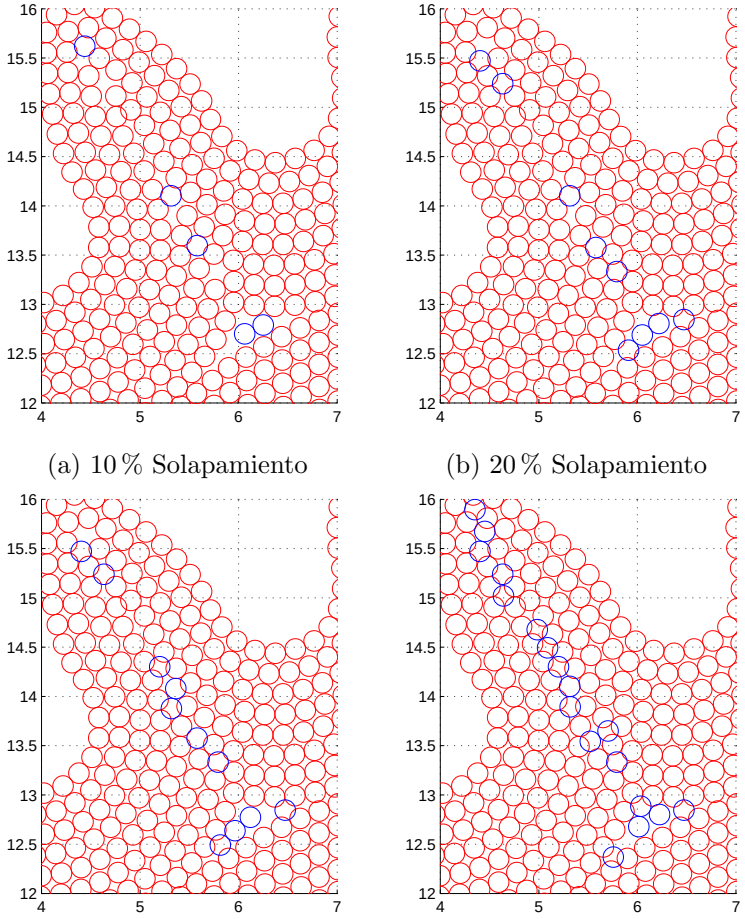


Figura D.7: Ejemplo de rellenado densp, simulado con MATLAB

D.6. Efecto de solapamiento



(a) 10 % Solapamiento

(b) 20 % Solapamiento

(c) 30 % Solapamiento

(d) 40 % Solapamiento

Figura D.8: Ejemplo con diferentes grados de solapamiento, simulado en MATLAB

Anexo E

Ejemplo de G-Code

```
// ***********ARBURG SLICER SETTINGS*************************
// Anzahl an STLs: 1
// ----------
// 1. geslicetes STL: C:\Data\STL\Planet.stl
// Verwendetes Material: M1
// Verhaeltnis B/H: 1.04
// Anzahl der Oberflaechenschichten: 0
// Fuellgrad der Oberflaechenschichten: 1.00
// Fuellgrad der Innenschichten: 1.00
// Anzahl der Konturen: 8
// Jitter: 10.00
// ueberlapp innerste Kontur mit Fuellung: 0.00
// Konturen von innen nach aussen gedruckt: true
// Konturen vor Fuellung: true
// Startwinkel der Fuellung: 45.0
// Versatzwinkel der Fuellung: 90.0
// Offset-Faktor der ersten Schichthoehe: 1.00
// Zusaetzlicher Offset-Faktor fuer aeussere Konturen: 0.00
// Zusaetzlicher Offset-Faktor fuer innere Konturen: 0.00
// ----------
// Slice Hoehe (global): 0.215
// ----------
// Materialien
// M1 ID: {02c2ef76-3b7f-4fc8-adcb-f55037f7e714}
// M1 Versions ID: {{}}
// ----------
T4S{M1}
G01 F48000 X11.9792 Y5.5037 Z0.2150
G01 F24000 X11.9792 Y5.5037 Z0.2150 T1
G01 F24000 X11.8291 Y5.3379 Z0.2150 T1
```

```
G01 F24000 X11.6730 Y5.1779 Z0.2150 T1
G01 F24000 X11.5005 Y5.0355 Z0.2150 T1
G01 F24000 X11.3264 Y4.8952 Z0.2150 T1
G01 F24000 X11.5500 Y4.8974 Z0.2150 T1
G01 F24000 X11.7720 Y4.9242 Z0.2150 T1
G01 F24000 X11.8450 Y5.1356 Z0.2150 T1
G01 F24000 X11.9132 Y5.3485 Z0.2150 T1
G01 F24000 X11.9792 Y5.5037 Z0.2150 T1
G01 F48000 X10.3989 Y4.3363 Z0.2150
G01 F24000 X10.3989 Y4.3363 Z0.2150 T1
G01 F24000 X10.1900 Y4.2565 Z0.2150 T1
G01 F24000 X9.9810 Y4.1769 Z0.2150 T1
G01 F24000 X9.7692 Y4.1053 Z0.2150 T1
G01 F24000 X9.5509 Y4.0569 Z0.2150 T1
G01 F24000 X9.7599 Y3.9774 Z0.2150 T1
G01 F24000 X9.9742 Y3.9134 Z0.2150 T1
G01 F24000 X10.1393 Y4.0642 Z0.2150 T1
G01 F24000 X10.2876 Y4.2315 Z0.2150 T1
G01 F24000 X10.3989 Y4.3363 Z0.2150 T1
G01 F48000 X8.5053 Y3.9307 Z0.2150
G01 F24000 X8.5053 Y3.9307 Z0.2150 T1
G01 F24000 X8.2817 Y3.9321 Z0.2150 T1
G01 F24000 X8.0593 Y3.9550 Z0.2150 T1
G01 F24000 X7.8369 Y3.9789 Z0.2150 T1
G01 F24000 X7.6153 Y4.0088 Z0.2150 T1
G01 F24000 X7.7767 Y3.8540 Z0.2150 T1
...
...
G01 F24000 X7.0441 Y1.1870 Z5.8050 T1
G01 F24000 X7.0921 Y0.9686 Z5.8050 T1
G01 F24000 X7.1546 Y0.7539 Z5.8050 T1
G01 F24000 X7.2297 Y0.5433 Z5.8050 T1
G01 F24000 X7.2888 Y0.4002 Z5.8050 T1
M30
```

Anexo F

Diagramas de proceso

En este anexo se presentan los diagramas de proceso específicos para cada módulo presentado en la sección 4.2.

F.1. Parser

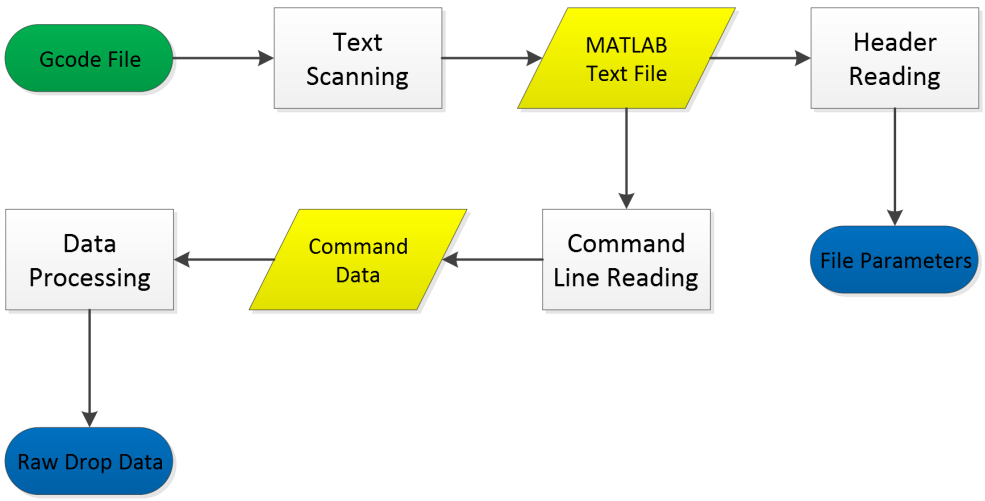


Figura F.1: Diagrama de flujo del Parser

F.2. Analizador

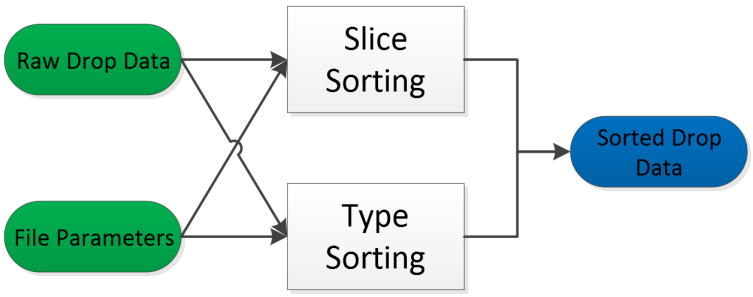


Figura F.2: Diagrama de flujo del Analyzer

F.3. Rellenado de vacíos

El parámetro de configuración en este caso es el grado de solapamiento elegido.

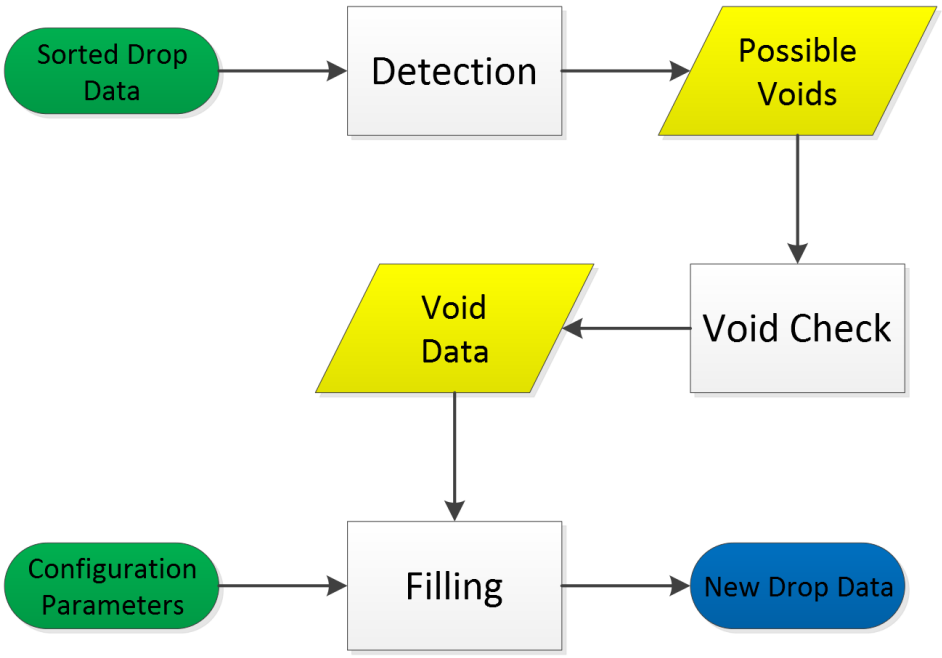


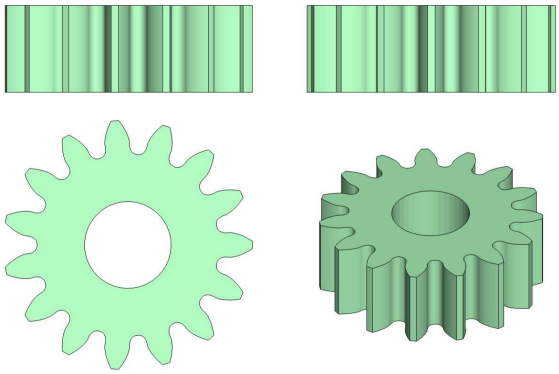
Figura F.3: Diagrama de flujo del rellenado de vacíos

Anexo G

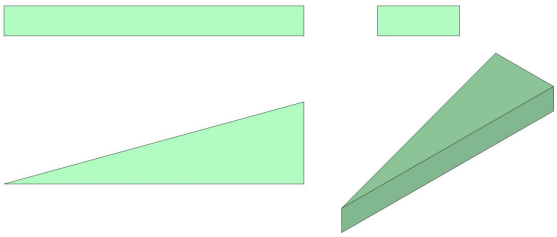
Configuración de los experimentos

En este anexo se presentan imagenes y esquemas sobre el equipamiento y configuración de los experimentos llevados a cabo.

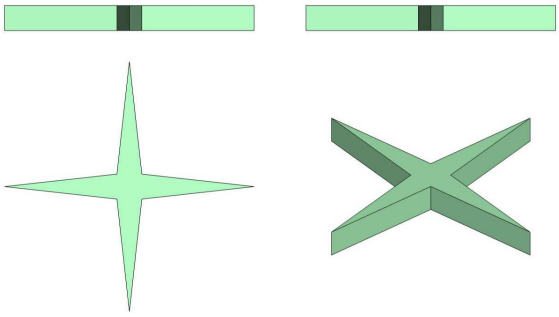
G.1. Modelos Simulados



(a) Modelo "Planet"



(b) Modelo "Spitz15"

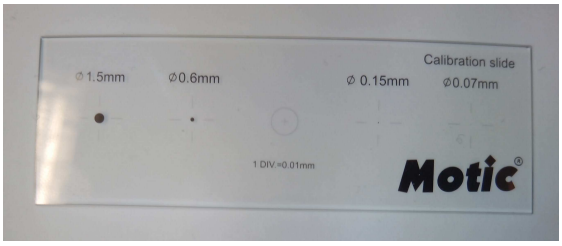


(c) Modelo "Stern2"

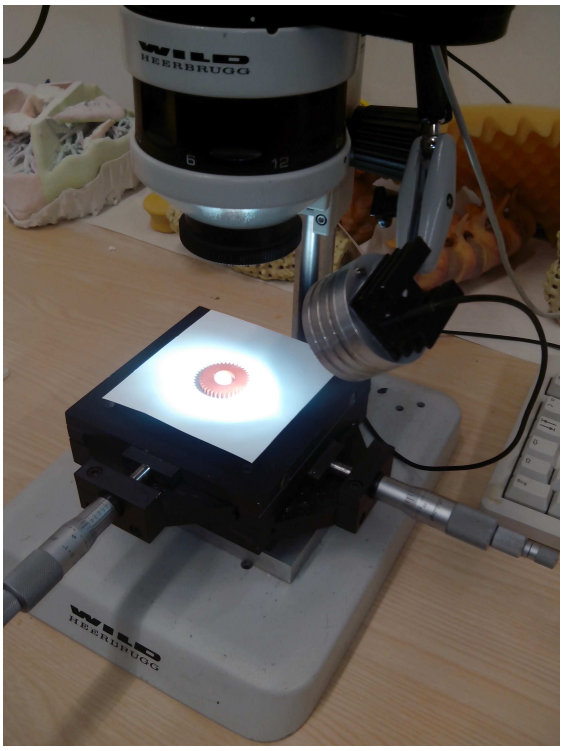
Figura G.1: Modelos usados en el test de MATLAB. La disposición de la imágenes es, de izquierda a derecha y de arriba a abajo: frontal, lateral, planta y perspectiva isométrica.

G.2. Microscopios y medición

El análisis bajo microscopio se realiza con el siguiente equipamiento:



(a) Placa de calibración



(b) Microscopio Wild Heerbrugg M5A

Figura G.2: Part of the equipment used in the experiment

El proceso de calibrado y medición es el siguiente:

1. Calibrado del microscopio: para obtener mediciones correctas, la cámara conectada al microscopio debe ser calibrada. Con la ayuda de la placa de calibración (figura G.2a) la cámara se calibra para una amplificación de 25x y un enfoque determinado.

2. Inspección: la pieza se coloca en el soporte del microscopop (figura G.2b) y se fija el objetivo sobre el vacío a medir.

3. Medición: se toma una imágen del vacío. El software de la cámara mide la superficie del vacío (figura G.3).

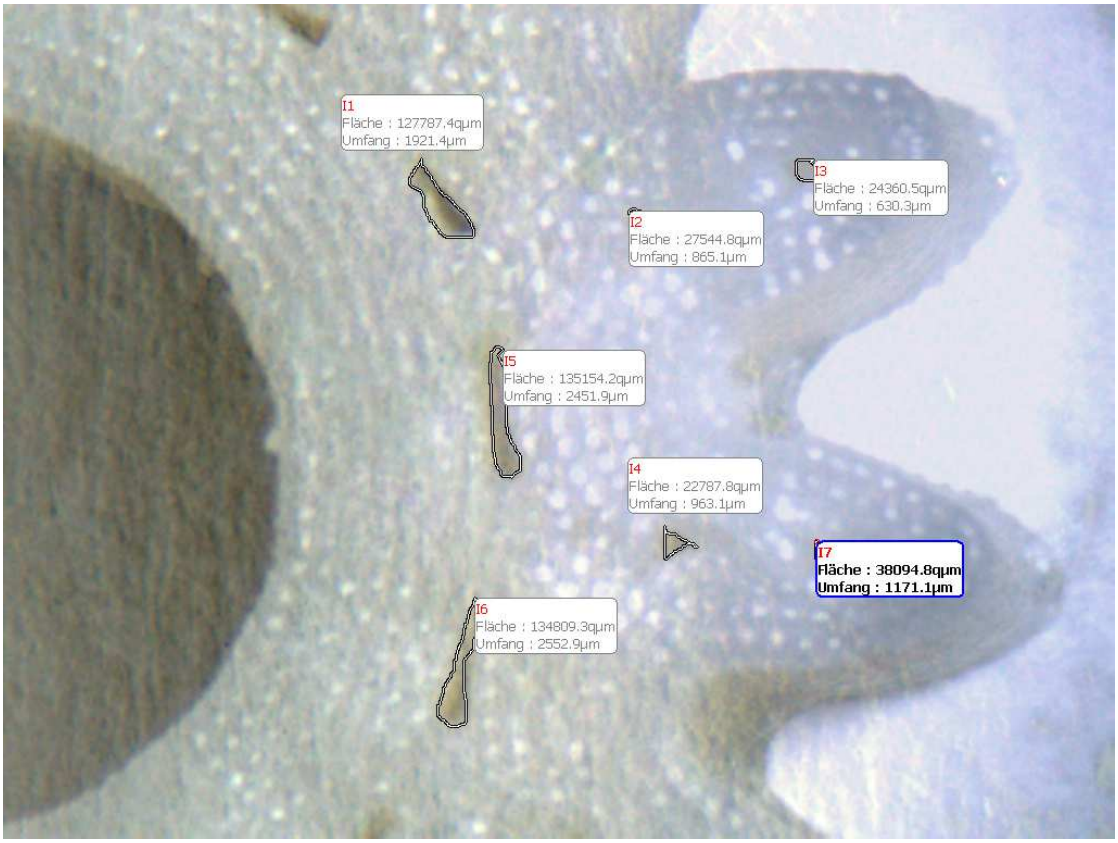


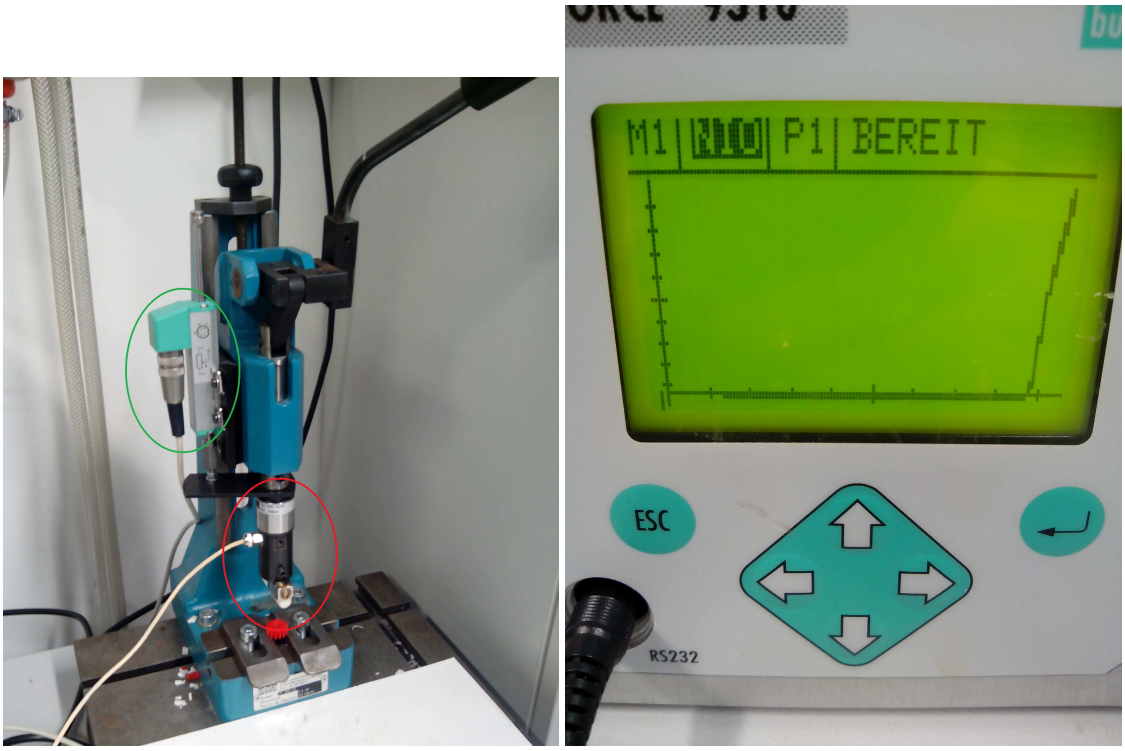
Figura G.3: Medición de la superficie

G.3. Rugosímetro

Figura G.4: Rugosímetro empleado, [Julian Müller, 2014]

G.4. Ensayo de flexión

El ensayo de flexión se realiza con el siguiente equipamiento:



(a) Pistón de presión, en verde el sensor de desplazamiento, en rojo el sensor de fuerza.

(b) Monitor que genera las curvas tensión-desplazamiento

Figura G.5: Equipamiento usado en el ensayo de flexión

El ensayo consiste en bajar el pistón, accionado manualmente hasta que la pieza rompe:

(a) Posición inical (b) Posición de rotura

Figura G.6: Proceso de ensayo

Anexo H

Tabla T de Student

Studentsche t–Verteilung

Die Formel der STUDENTschen t–Verteilung s. 16.2.4.8, S. 786.

STUDENTsche t-Verteilung: Quantile $t_{\alpha,m}$ bzw. $t_{\alpha/2,m}$

Anzahl der Freiheitsgrade	Wahrscheinlichkeit α für zweiseitige Fragestellung					
m	0,10	0,05	0,02	0,01	0,002	0,001
1	6,31	12,7	31,82	63,7	318,3	637,0
2	2,92	4,30	6,97	9,92	22,33	31,6
3	2,35	3,18	4,54	5,84	10,22	12,9
4	2,13	2,78	3,75	4,60	7,17	8,61
5	2,01	2,57	3,37	4,03	5,89	6,86
6	1,94	2,45	3,14	3,71	5,21	5,96
7	1,89	2,36	3,00	3,50	4,79	5,40
8	1,86	2,31	2,90	3,36	4,50	5,04
9	1,83	2,26	2,82	3,25	4,30	4,78
10	1,81	2,23	2,76	3,17	4,14	4,59
11	1,80	2,20	2,72	3,11	4,03	4,44
12	1,78	2,18	2,68	3,05	3,93	4,32
13	1,77	2,16	2,65	3,01	3,85	4,22
14	1,76	2,14	2,62	2,98	3,79	4,14
15	1,75	2,13	2,60	2,95	3,73	4,07
16	1,75	2,12	2,58	2,92	3,69	4,01
17	1,74	2,11	2,57	2,90	3,65	3,96
18	1,73	2,10	2,55	2,88	3,61	3,92
19	1,73	2,09	2,54	2,86	3,58	3,88
20	1,73	2,09	2,53	2,85	3,55	3,85
21	1,72	2,08	2,52	2,83	3,53	3,82
22	1,72	2,07	2,51	2,82	3,51	3,79
23	1,71	2,07	2,50	2,81	3,49	3,77
24	1,71	2,06	2,49	2,80	3,47	3,74
25	1,71	2,06	2,49	2,79	3,45	3,72
26	1,71	2,06	2,48	2,78	3,44	3,71
27	1,71	2,05	2,47	2,77	3,42	3,69
28	1,70	2,05	2,46	2,76	3,40	3,66
29	1,70	2,05	2,46	2,76	3,40	3,66
30	1,70	2,04	2,46	2,75	3,39	3,65
40	1,68	2,02	2,42	2,70	3,31	3,55
60	1,67	2,00	2,39	2,66	3,23	3,46
120	1,66	1,98	2,36	2,62	3,17	3,37
∞	1,64	1,96	2,33	2,58	3,09	3,29
	0,05	0,025	0,01	0,005	0,001	0,0005
	Wahrscheinlichkeit α für einseitige Fragestellung					

Figura H.1: Tabla T de Student utilizada en los cálculos

Anexo I

Resultados de los experimentos

En este anexo se presentan las tablas de resultados numéricos de los distintos experimentos.

I.1. Tablas test MATLAB

Cuadro I.1: Resultados de la simulación con rellenado por contornos

Contornos	**Volumen Inicial** $[mm^3]$	**Grado de Solapamiento** [%]	**Nuevo Volumen Añadido** $[mm^3]$	**Volumen Solapado Añadido** $[mm^3]$	**Solapamiento** [%]
Planet	465,08	10	11,24	0,18	1,58
		20	19,16	1,17	6,11
		30	27,03	3,34	12,36
		40	38,09	7,70	20,22
		50	227,70	99,04	43,49
Spitz15	922,86	10	3,83	0,03	0,73
		20	9,72	0,56	5,76
		30	15,01	1,98	13,22
		40	18,27	3,38	18,51
		50	292,02	133,31	45,65
Stern2	310,21	10	6,20	0,09	1,44
		20	10,27	0,58	5,66
		30	12,60	1,16	9,18
		40	16,38	2,56	15,60
		50	133,46	59,46	44,56

Cuadro I.2: Resultados de la simulación con rellenado por mallado

Mallado	**Volumen Inicial** $[mm^3]$	**Grado de Solapamiento** [%]	**Nuevo Volumen Añadido** $[mm^3]$	**Volumen Solapado Añadido** $[mm^3]$	**Solapamiento** [%]
Planet	498,17	10	-	-	-
		20	0,11	0,02	18,70
		30	2,24	0,57	25,36
		40	7,10	2,39	33,73
		50	157,65	75,09	47,63
Spitz15	938,85	10	0,26	0,01	2,16
		20	2,24	0,31	13,86
		30	7,66	1,62	21,18
		40	12,34	3,53	28,63
		50	395,83	185,66	46,90
Stern2	322,34	10	0,61	0,01	1,22
		20	2,43	0,26	10,72
		30	6,57	1,29	19,68
		40	10,48	2,87	27,43
		50	119,55	55,46	46,39

Cuadro I.3: Resultados de la simulación con rellenado denso

Denso	**Volumen Inicial** $[mm^3]$	**Grado de Solapamiento** [%]	**Nuevo Volumen Añadido** $[mm^3]$	**Volumen Solapado Añadido** $[mm^3]$	**Solapamiento** [%]
Planet	498,17	10	0,07	0,00	1,44
		20	7,55	1,01	13,33
		30	18,07	3,85	21,32
		40	27,69	7,62	27,52
		50	70,42	28,37	40,29
Spitz15	929,08	10	0,30	0,01	2,26
		20	3,72	0,47	12,60
		30	11,37	2,57	22,59
		40	21,16	6,12	28,92
		50	48,44	18,74	38,68
Stern2	315,76	10	0,66	0,01	1,37
		20	4,01	0,44	10,94
		30	10,30	2,03	19,70
		40	16,15	4,20	25,99
		50	45,78	18,84	41,15

I.2. Tabla de medidas de rugosidad

Cuadro I.4: Mediciones del ensayo de rugosidad

Rugosidad [μm]	
Algoritmo	
Sin	Con
6,065	7,889
5,294	5,026
5,61	5,449
7,391	5,307
4,871	3,847
5,354	4,234
7,503	5,646
7,675	4,344
4,257	4,01
4,162	3,472
6,467	3,476
7,512	4,462
7,181	3,838
6,198	4,593
4,174	3,971
3,921	
7,095	
5,183	

I.3. Tabla de ensayo de flexión

Cuadro I.5: Tabla de tensiones de fractura

	Tensión de fracura $[kN]$			
	Contornos		Denso	
	Algoritmo		Algoritmo	
	Sin	Con	Sin	Con
	0,685	0,751	1,05	1,002
	0,67	0,821	0,98	0,873
	0,663	0,766	0,947	0,987
	0,652	0,67	0,851	0,991
	0,729	0,707	0,954	1,098
	0,667	0,748	0,961	1,028
	0,667	0,711	0,987	1,031
	0,718	0,733	0,902	0,851
	0,715			1,09
Media	0,685	0,738	0,954	0,995
Des. Típica	0,028	0,045	0,059	0,085