

COB-2023-0685

RP3: A RESEARCH FRAMEWORK FOR PROCESS PLANNING IN ADDITIVE MANUFACTURING

Neri Volpato

Felipe Gonçalves Di Nisio

Federal University of Technology - Paraná (UTFPR), Department of Mechanical Engineering, Additive Manufacturing and Tooling Group (NUFER), Curitiba-PR, Brazil

nvlpato@utfpr.edu.br, felipenisio@alunos.utfpr.edu.br

Henrique Romaniuk Ramalho

Rodrigo Minetto

Ricardo Dutra da Silva

Federal University of Technology - Paraná (UTFPR), Department of Informatics, NUFER, Curitiba-PR, Brazil

henriqueramalho@alunos.utfpr.edu.br, rminetto@utfpr.edu.br, rdsilva@utfpr.edu.br

Abstract. *To use Additive Manufacturing (AM) to its full potential, besides an expert user, a good Computer-Aided Manufacturing (CAM) tool for process planning is paramount. The main AM process planning steps are: build setup, responsible for defining orientation, positioning, and scaling of the 3D model; slicing; support calculation; filling and machine data generation. Several decisions that will affect cost, quality, and manufacturing time need to be taken in this process. The AM and tooling group (NUFER) has been developing a CAM for AM called Rapid Prototyping Process Planning (RP3) for over a decade, and it is now in its third generation. The main objective is to develop research to improve AM process planning. RP3 can work with the main AM categories and offers complete control of the methods to perform the process planning tasks. This paper presents the main features and some of the late implementation of RP3. The highlighted contributions are an optimal slicing method, mesh topology, tool-path optimization, and many filling strategies to generate porous structures. As it has been developed as a modular, easy-to-adapt software, different methods for the same task can be designed and easily integrated, becoming an ideal framework for research and development in this area.*

Keywords: 3D printing, Slicing, Filling, Support, Algorithm.

1. INTRODUCTION

Additive Manufacturing (AM), also known as 3D printing, is already part of the daily activities of many companies and areas, having a fundamental role in prototyping, tooling and, more recently, in the production of final parts. Both technologies and materials have improved considerably since the early days, and these, together with the great potential offered by the additive principle, have contributed to consolidating AM as an important manufacturing process. In addition to having an expert user, an efficient AM process planning tool, also known as a Computer-Aided Manufacturing (CAM) for AM, is required to be successful with these technologies. The CAM for AM is popularly called a *slicer* , but as it can be observed below, this term does not reflect all necessary tasks required in the process planning.

The general AM process chain starts with the 3D geometric model of the part, which goes through the process planning, the manufacturing stage, and the post-processing to finish the final part (see Figure 1). The ISO/ASTM 52900:2015 sorts the AM technologies in seven categories: material extrusion (MEX), vat photopolymerization (VPP), material jetting (MJT), binder jetting (BJT), sheet lamination (SHL), powder bed fusion (PBF), and directed energy deposition (DED). They all share similar process planning steps, also highlighted in Figure 1: the build setup, responsible for defining orientation, positioning, and scaling of the 3D model; slicing; support generation (if required); filling or path-planning; and machine code generation.

In each of these steps, some decisions need to be taken that will affect many aspects of the final part, including cost, quality, and manufacturing time. A good process planning tool can help users to achieve the best of the AM. In the early days, and even today, the AM companies developed their dedicated CAM software, not accepting machine code from any other system. However, with the expiration of the first AM patents, many new companies entered this market and decided not to use dedicated CAM software. This decision helped to speed up the developments in the area, and many methods have been proposed in the literature to improve the process concerning CPU performance, build time, part strength, accuracy, surface finishing, and other aspects. Today, there are some free or even open software to perform the process planning. However, they are usually limited to the MEX technology and were not developed to easily allow the implementation of new methods.

The additive manufacturing and tooling group (NUFER) at the Federal University of Technology - Paraná (UTFPR)

has been developing a CAM for AM, called Rapid Prototyping Process Planning (RP3), for over a decade (Volpato and Foggatto, 2009), and it is now in its third generation (V3.0). The group’s main objective is to conduct research in process planning to improve many aspects of AM, and RP3 can help researchers to develop new methods in any step of the process planning and, in theory, in any AM category.

This paper presents the main features and some of the late implementations of the RP3 V3.0. The highlighted contributions are mesh topology, tool-path optimization, and filling strategies to generate porous structures.

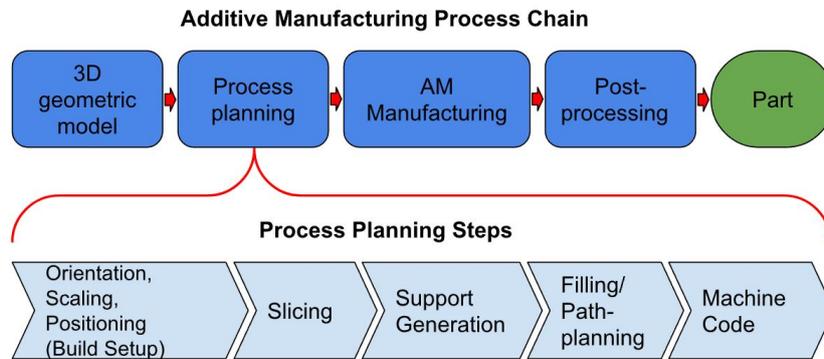


Figure 1: Additive Manufacturing process chain and process planning steps.

2. PROCESS PLANNING FRAMEWORK FOR AM

The RP3 is a generic AM process planning system and evolved considerably from its previous version (Volpato and Foggatto, 2009). Currently, it covers four different AM technologies, namely: MEX, VPP, PBF (laser-based PBF, L-PBF), and DED, which are schematically presented in Figure 2. The goal is to allow research in process planning (developing new methods, strategies, optimizations, etc.), offering greater autonomy to control ordinary and new process parameters that any specific method may require. The RP3 is developed in C++, following the Object-Oriented Programming (OOP) paradigm. The following sections describe the software structure and the main implementations.

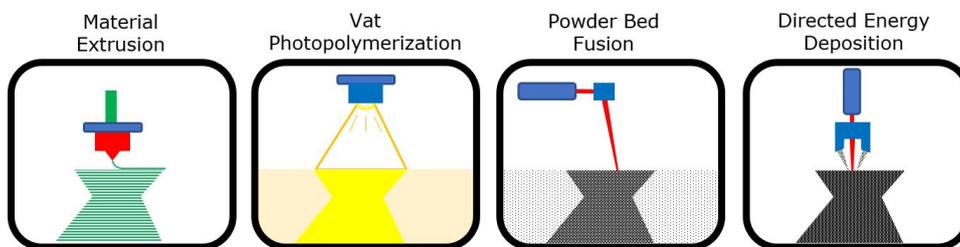


Figure 2: AM categories already implemented in RP3.

2.1 Modular structure

RP3 is structured with a modular concept, where each process planning step can be executed using different methods (algorithms). To achieve that, for each step, the input and output data were defined as presented in Figure 3. In this way, a developer knows exactly what is received from a precedent step and what should be delivered to the next one. By doing this, a framework for research and development in this area was created, allowing different methods for the same step to be plugged in. This approach offers complete control of all algorithms to perform the process planning tasks. Figure 3 also presents the methods already implemented in the framework, some currently being developed, and some intended to be implemented in the future. Some of these implementations are described in the following sections.

In the RP3 framework, the steps involving build setup (orientation, scaling, and positioning) and slicing were considered *generic*, i.e., they are almost the same for all AM categories (see Figure 3). This means that their input and output (3D models, build geometries and the pile of layers) do not change. Strictly speaking, as pointed out below, the part positioning in the build (packing) could not be considered a generic task because it has some restrictions based on the AM technology used, but it was kept like that for simplicity. The support generation, path-planning, and machine code generation are considered specific steps because the required data vary considerably for each technology. For instance, L-PBF of polymers and BJT do not require support structures, and, to add material in each layer, MEX requires tool-path and VPP image data. Then, the output and input (support and part slices, tool-path, images) in these steps must be specific.

In summary, the framework works with the following data structures. Firstly, a 3D model is stored by a *Mesh* type object, which represents a part in the 3D space. After the slicing step, for each part of the build, a *Solid* is instantiated. The *Solid* is responsible for storing a list of *Slice* objects (layers). Each *Slice* is made up of lists of *Contour* objects, which are 2D contours related to the boundaries of 3D models or support regions. In addition, a *Slice* also stores sequences of *Point* objects defining filling paths.

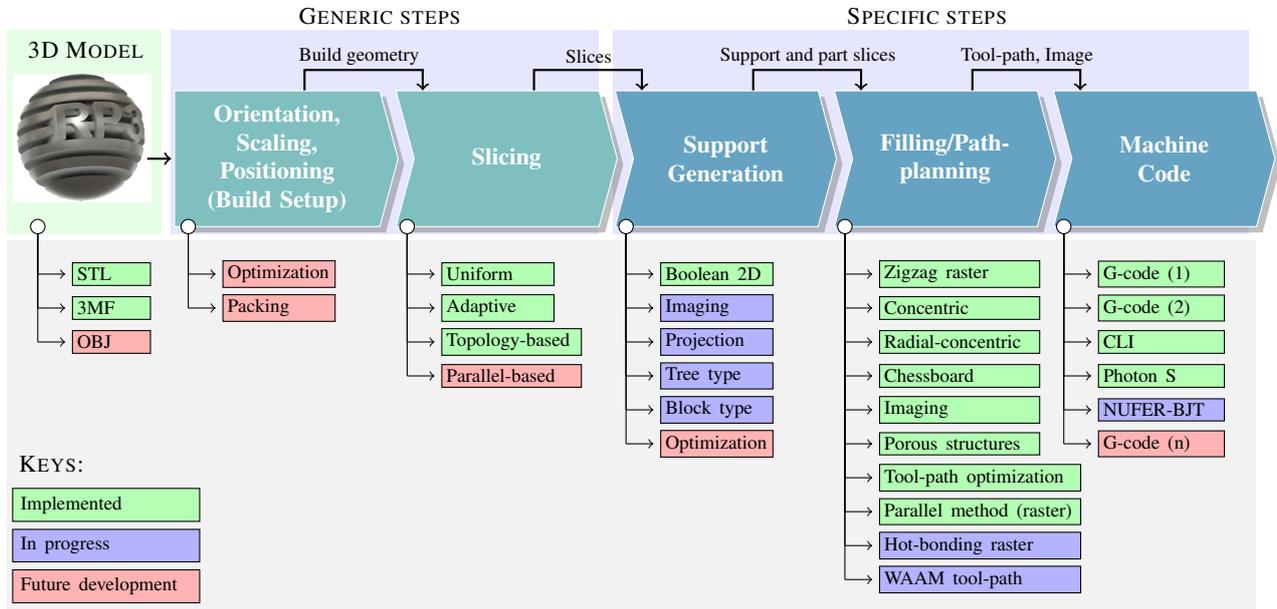


Figure 3: RP3 process planning framework.

The input for any slicing strategy is a list of *Mesh* objects (Build geometry), while the output is the list *Slice* (composed of *Contour* objects). For the support generation strategies, the input is the *Slice* list (with the *Contour* objects), and the output consists of new lists of *Contour* objects representing support regions, that are added to the *Slice* objects to compose each layer. Finally, the input for the filling step is the list of *Slice* objects with *Contour* objects related to part and support regions. The filling calculation is carried out with each *Contour*, and the output is a sequence of *Point* objects defining filling paths. For technologies such as VPP, the contours and filling paths are converted to images as outputs.

After the filling step, all *Solid* objects are merged into a single *Printing Solid* object. In this way, the route optimization process is facilitated since the *Printing Solid* object encompasses the contours and printing trajectories of all geometries in the build. Machine code generation depends on the AM technology and also on the specific machine employed. In general, for MEX, L-PBF, and DED, the list of points defining the paths for a layer of the *Printing Solid* is post-processed to generate the machine code, and for VPP, all *Contour* objects (part and support) are converted to images in the adequate output format.

2.2 3D Geometric model and Job setup

The initial input data, corresponding to the 3D models, are given in STL (STereoLithography) format (binary or ASCII) or as 3MF (3D Manufacturing Format). If required, the framework is ready to incorporate other formats, such as OBJ (Object File) and AMF (Additive Manufacturing File). Once the 3D models are selected, a list of *Mesh* objects is created containing the representation of each model.

Until the previous version of RP3, after reading an STL file, the 3D model was represented as a non-topological data structure. The data structure basically reflected the STL format, i.e., a list of triangles (often called a triangle soup) defined by three vertex coordinates. This type of representation has its drawbacks, most notably: (i) a vertex shared by triangles is replicated at each triangle, incurring in redundant data stored in computer memory; (ii) the search for matching vertices can be inaccurate due to numerical imprecision; and (iii) the lack of adjacency information makes it costly to search for triangles sharing vertices and edges (Lee and Woo, 2000).

Currently, RP3 can work both with a non-topological data structure (list of triangles) and a topological data structure (Half-Edge (Berg *et al.*, 2008)). Traditional algorithms for slicing and support computation rely on non-topological data structures. Adding the topological data structure to RP3 opened the opportunity to explore different process planning methods. For instance, slicing and support computation steps are now being developed to avoid the drawbacks mentioned above of triangle soups.

In order to initiate the process a *Job* is created, which configures the build setup containing all parts to be manufactured. The *Job* requires some information related to the technology and the machine (platform's geometry and volume);

therefore, a predefined machine database is presented to the user. RP3 allows the creation of more than one job in the same section or even jobs for different AM technologies, which are useful for comparing manufacturing strategies and technologies.

The build setup is considered a fundamental step in process planning. Orientation is one of the main decisions as it affects part accuracy, surface finishing, build time and cost, support structures, mechanical properties, etc. Part position in the platform is also relevant to reduce part cost because when the objective is production, it is directly related to the number of parts in a batch. Although it is here considered a generic step, part position might be restricted to a 2D or 3D nesting, according to the AM technology. Lastly, scaling is a simple geometric operation required by some technologies to compensate for shrinkage or other post-processing demands. Identifying of the correct scale value might be a laborious experimental task, however.

Although build setup is paramount, it is the less explored process planning step in RP3. Today, no optimized orientation or positioning methods have been implemented. This means the users must rely on their experience to perform these tasks.

Finally, each part's attributes (orientation, position and scaling) defined in this process are applied directly to their respective geometry representations (*Mesh* objects). It is worth emphasizing that RP3 was designed to allow setting each part in a *Job* with specific process parameters. Therefore, the parameters required in the following steps are associated with each individual *Mesh* object. The framework output of this first step is the set of geometries in a build ready for the slicing step.

2.3 Slicing

RP3 works both with uniform or adaptive slicing (Figure 4a). As each part in a *Job* can have specific process parameters, it is possible to slice each part in the build with a specific layer thickness or even apply uniform and adaptive slicing in the same build.

The input of this stage is the list of meshes (Build geometry) constructed before in the build setup. Given a *Mesh*, the framework retrieves all specified parameters for the geometry and passes them to the slicing routine. The slicing method is implemented according to an asymptotically optimal algorithm proposed by Minetto *et al.* (2017). The method iterates through the triangles of a mesh in a bottom-up strategy, maintaining at each slicing plane a set of active triangles intersected by it. In order to maintain the active list of triangles, the method organizes the triangles by their minimum *Z* coordinate (Figure 4b). Whenever the minimum coordinate of a triangle is reached, it is included in the list of active triangles. A triangle is removed from the active list when the current slicing plane exceeds its maximum *Z* coordinate.

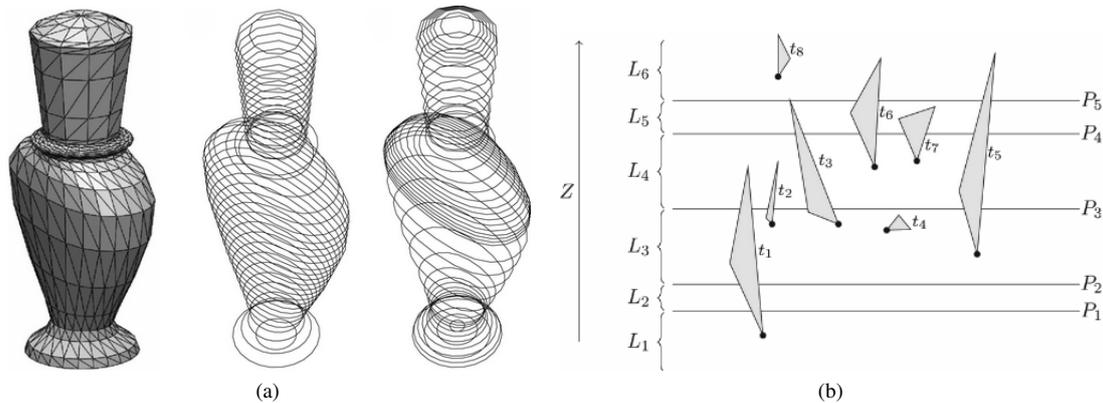


Figure 4: Uniform and adaptive slicing (a), and an example of triangle grouping by Z_{min} (b). Source: Minetto *et al.* (2017).

One of the slicing sub-tasks is contour construction, where the slicing segments are assembled into a close polygon. When mesh topology is used, obtaining a closed polygon is straightforward once each facet knows its neighbors. When the mesh topology can not be built, this sub-task must be performed by searching for the closest point from a list, which can be costly and not so robust. Another relevant sub-task is to identify the directions (clockwise or counter-clockwise) of the set of 2D closed contours (Volpato *et al.*, 2013). This information is necessary because, based on these directions, the filling algorithm knows if a contour must be filled (has material) or not (it is an empty area).

For each geometry in the build, a *Solid* is instantiated at first, and a list of *Slice* objects is attached to it. It is worth noting that the amount of layers is defined by the slicing thickness and the piece's height. The output of this step is a list of *Slice* objects, each with a list of *Contour* objects related to the boundaries of the 3D models.

2.4 Support calculation

This step has as input the list of *Slice* objects of each *Solid*. Currently, the implemented support generation method is based on Boolean operations over 2D polygons (Schneider and Eberly, 2002; Chen and McMains, 2005). In this, the support region is identified for each layer using the polygons of adjacent layers. Given any layer L_i and the layer immediately above it L_{i+1} , the difference of polygons in L_{i+1} and L_i represent suspended regions of L_{i+1} . This operation can identify all hanging regions independently of their inclination, as seen in Figure 5a and 5b. If the AM technology has a self-support angle (above which no support structure is required), the unnecessary regions need to be removed, as in Figure 5c. To do that, the difference between layers is performed using a dilated version of the polygons in L_i . The dilation factor δ is a simple relation of the self-support angle θ and the distance $d(i, i + 1)$ between the two adjacent layers, i.e. $\delta = d(i, i + 1) / \tan \theta$. These Boolean methods are also being adapted to work using images (Huang *et al.*, 2014). This option is adequate for VPP technologies because the filling step is already performed using images.

Another support method under development is based on the projection of all negative facets of the model (related to the Z direction), which are below the self-support angle. If the facets normal are correct, the projection method can facilitate the identification of the support regions once less calculation is required.

The support structures can be created once the support regions have been identified. Currently, basic zigzag and block-like filling structures have been implemented for MEX and L-PBF. Some new support options, such as columns and tree-like structures, are under development to apply less costly structures (Figure 5d), but they still need to be fully implemented. Some AM technologies use a base between the part and the machine platform, obtained using the same Boolean methods described above.

The output of this step are lists of *Contour* objects, related to the support regions that are attached to their corresponding *Slice* in the layer sequence.

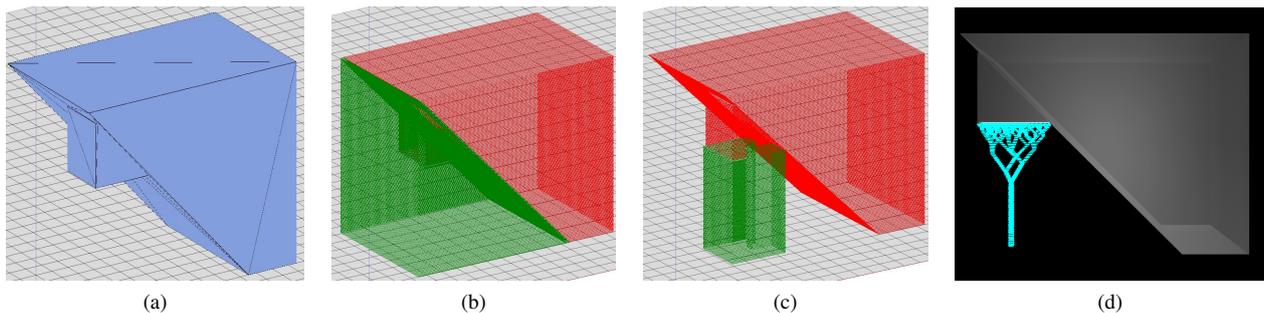


Figure 5: Support strategies: (a) 3D model; (b) all hanging region; (c) considering self-supported regions; and (d) tree-based support.

2.5 Filling or path-planning

The filling or path-planning step takes the list of *Slice* objects of each part in the build as input. This step is where the majority of RP3 developments were focused (see also Figure 3). As mentioned before, the RP3 permits to set specific process parameters for each part in a build, giving the freedom to test a variety of parameters in a single build. Additionally, some process parameters can be changed for each specific layer or in a layer range option, which means that the filling of each part can be tailored to a specific application or to yield a particular mechanical property. An idea of this potential was presented in the sandwich filling configuration described in Magalhães *et al.* (2014).

The current version of RP3 has several filling strategies for the AM technologies supported (Figure 6 presents an overview of them). Two of the most common AM filling patterns, zigzag/raster (rectilinear) and concentric filling strategies, are available for MEX, L-PBF, and DED. In both strategies, the user can choose to set an infill density (percentage) or to define a gap (or hatch distance) value. In the raster filling strategy, the user has control of all basic printing parameters, such as the raster angle and the rotation between layers.

For the MEX technology, the filling can be executed sequentially or in a parallel model using a Graphics Processing Unit (GPU) architecture. The GPU method is essential when it is necessary to analyze, for instance, the best raster angle to generate the tool-path, which requires a significant amount of computing. For instance, this method can reduce the manufacturing time by running concurrent simulations to find optimal raster angles that maximize the average length of raster lines (Faust *et al.*, 2023).

Another research field being studied is the tool-path optimization for MEX. The aim is to reduce jump distances (or airtime motions). Today, the optimization algorithms implemented are the greedy and the combination of Nearest Insertion and 2-Opt heuristics (Volpato *et al.*, 2020). The methods follow a data clustering and collapsing preprocessing approach proposed by Weller *et al.* (2021). The main idea is that once the tool-path of the whole build is calculated (obtaining the

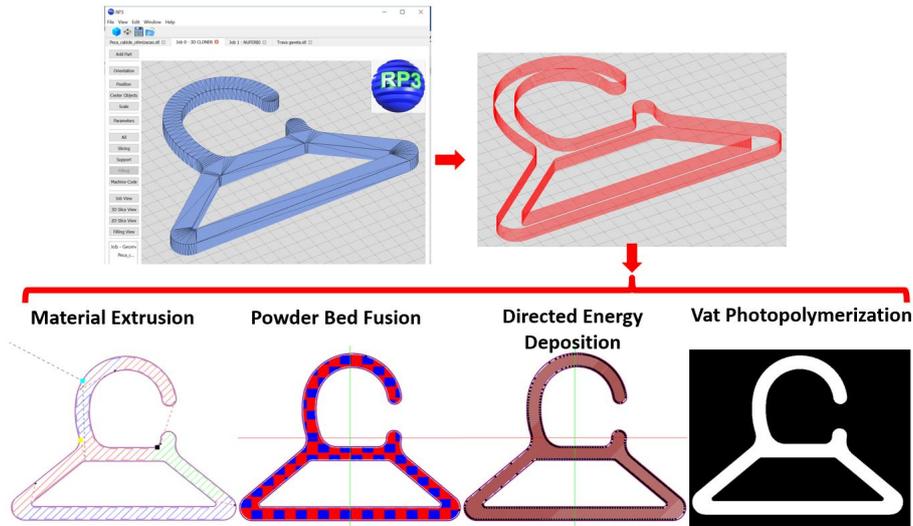


Figure 6: Some RP3 filling strategies according to the AM technology.

Printing Solid), the optimization algorithms search for a good sequence of contours and raster filling segments, which will minimize the jump distances to deposit material.

Some filling proposals are also being developed to strengthen the adhesion of adjacent material beads in the same layer in MEX, which might improve part strength. Recently, a cooling time constrained raster-fill planning algorithm was proposed to reduce the temperature gradient between successive raster filling segments. The method was designed to keep the cooling time intervals between depositions below user-specified limits (Nakonetchnei *et al.*, 2022).

For L-PBF and DED technologies, besides the basic raster strategy, the chessboard filling is also available (Laskoski *et al.*, 2021). This strategy divides the filling area into islands with shorter raster lengths and 90° rotation in the raster angle between islands (Figure 7a). This strategy is applied to promote lower temperature gradients between regions of the printed part, reducing surface tensions that can cause cracks or deformations during the fabrication of metallic parts. In RP3, it is also possible to set an XY displacement of islands between layers to reduce porosity between islands (Machado *et al.*, 2023). Figure 7b shows a part printed in an Alkimat Laser Funde 200 machine.

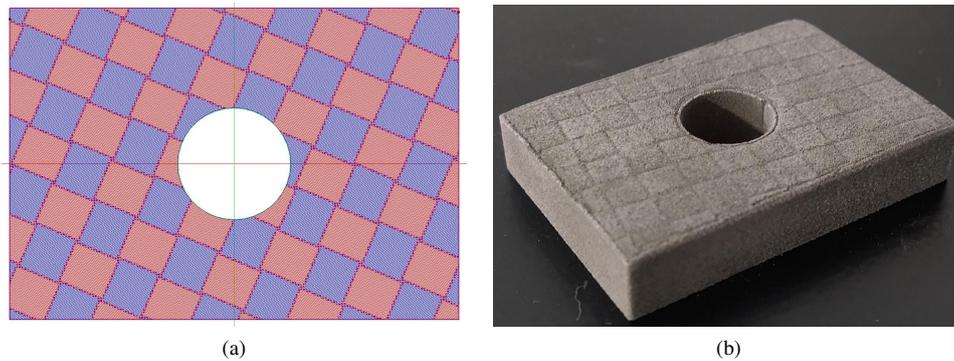


Figure 7: Chessboard strategy for L-PBF and DED (a) and printed part in an Alkimat Laser Funde 200 machine (b).

One particular application where the filling can contribute considerably is in porous media fabrication. Some advanced features and strategies were developed to this end. One of them, named *Joined Filaments*, enables printing two or more filaments side-by-side in *sets*, enabling to control also the distance between filaments and the distance between sets (Figure 8a) (Shigueoka and Volpato, 2022). There is also the *Stagger Distance* parameter, which applies a lateral displacement in the raster lines between layers with the same raster angle without changing the gap value (Figure 8b). Another strategy is called *Variable Road Gap*, which enables the user to define a piece-wise function to control a variable gap value on the layer (Figure 8c and 8d). Figure 8e shows the micro-computed tomography image of a porous structure's cross-section obtained by applying these advanced strategies in different parameter configurations distributed through the sample's layers.

Another type of porous structure for MEX can be obtained with the *Variable Bead Width Porous Filling* (VBWPF) strategy (Di Nisio *et al.*, 2022). In this case, filaments are deposited with variable widths in a zigzag pattern, creating spaces (pores) between adjacent filaments (Figure 9). The width variation is obtained by dividing the raster lines into sections with controlled length, printing speed, and extrusion rate. Figure 9c shows images of layers printed with

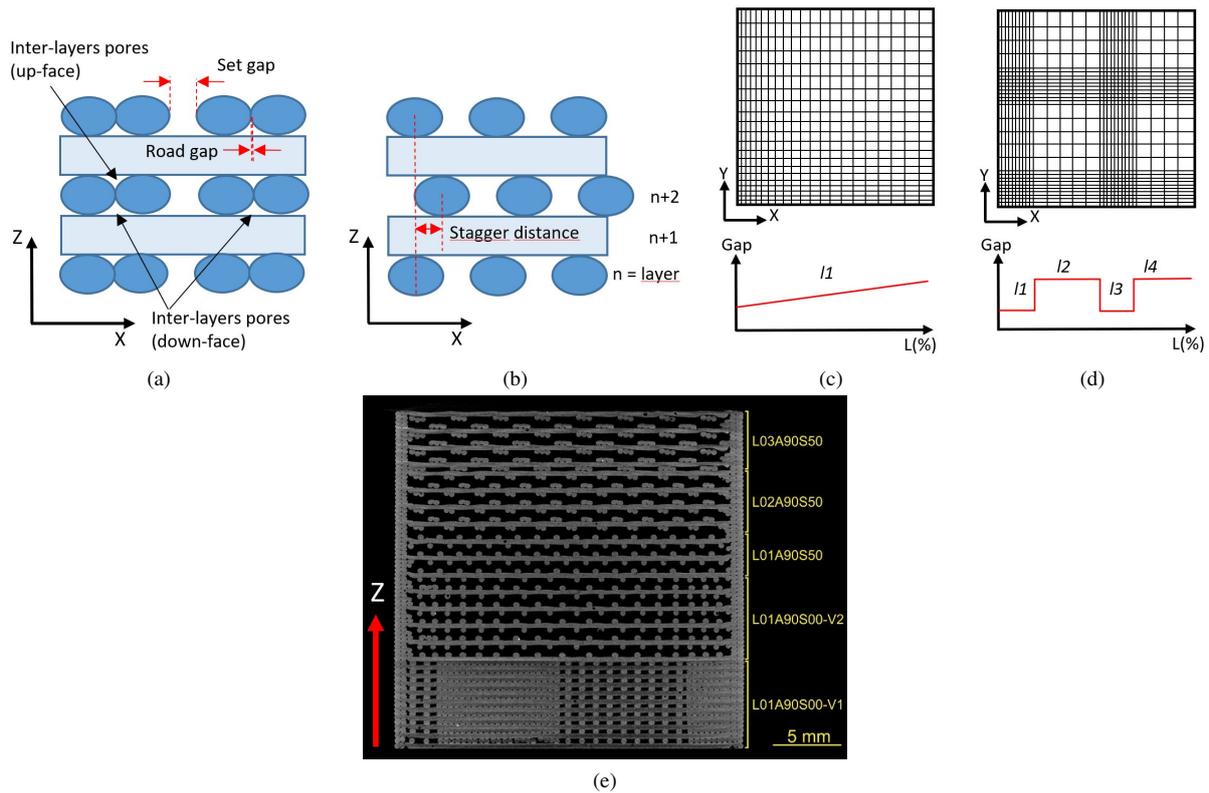


Figure 8: Advanced raster filling: *Joined Filaments* - sets of two filaments (a); *Stagger Distance* (b); and *Variable Road Gap*, applying a linear (c) and a piece-wise function (d); Micro-computed tomography cross-section of a porous structure with different advanced parameters configurations (e). Source: Shigueoka and Volpato (2022).

two different VBWPF configurations. The resulting porous structure has pores with higher aspect ratios and less pore interconnectivity compared to those obtained with traditional raster filling (Di Nisio *et al.*, 2022).

All these advanced parameters allowed more control of a part's internal geometry, enabling the user to design porous structures according to their needs.

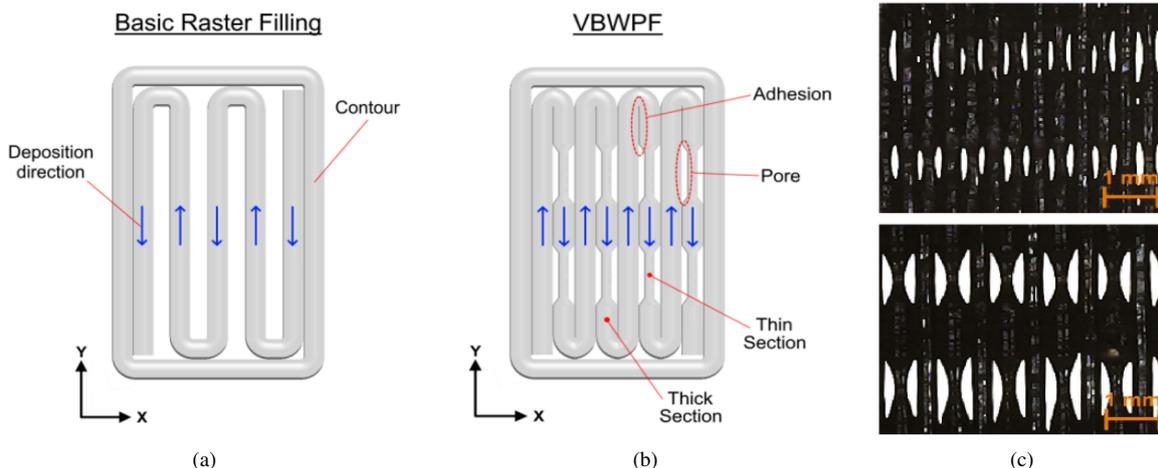


Figure 9: Difference between traditional raster filling (a) and the VBWPF strategy (b) and two examples of layers printed with different VBWPF configurations (c). Source: Di Nisio *et al.* (2022).

In the VPP technology, it is possible to use the default filling, which generates 2D images from the pile of slices to be printed. Another possibility developed for porous structures is to use the *Cell Porous Media* option, which can repeat in *X* and *Y* a *Unit cell* in a volume of a 3D model, defined as the *Target model*. In this case, two 3D models are loaded in the program: the unit cell and the target one (Figure 10a). The operation can be made by Boolean addition (example in Figure 10b) or subtraction. This feature enables the fabrication of complex cellular structures without the need to model them in Computer-Aided Design (CAD) software (Puppi *et al.*, 2022).

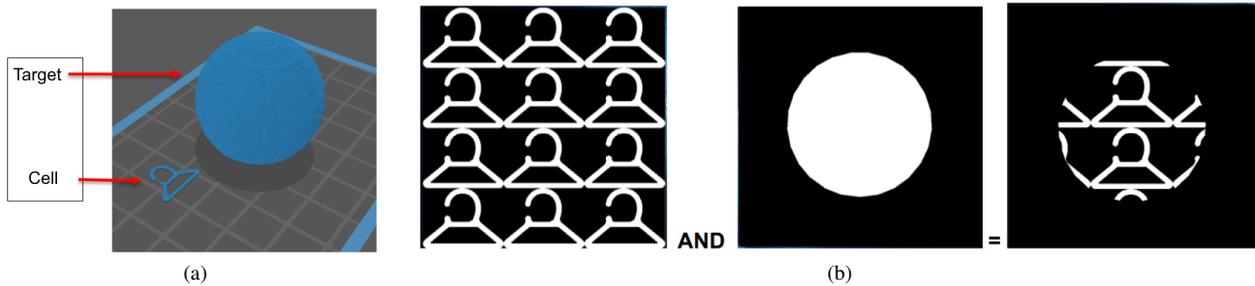


Figure 10: Example of a Boolean addition in the Cell Porous Media function for VPP technologies: target and cell 3D models (a) and operation *AND* between two images for a single layer (b). Source: Puppi *et al.* (2022).

In general, the outputs of all filling strategies are lists of *Point* objects that define the printing paths of each *Slice*. These paths are mapped to images for technologies such as VPP.

2.6 Machine code (post-processing)

The filling data structure (*Printing Solid* object) can be post-processed to any machine code. Today, the g-code is implemented for MEX polymer (Ender 5 printer) and ceramic (delta Duraprinter), and also for DED LMD technology from UFSC (Volpato *et al.*, 2019) and SENAI-Joinville. For L-PBF, the Common Layer Interface (CLI) is used to control the process of the Alkimat Laser Funde 200 machine. The photon S format is implemented for a VPP printer (Photon S LCD). The group is currently developing its own BJT output, called NUFER-BJT, for a research apparatus under development (Silvy *et al.*, 2021). In theory, if the machine data format is known, RP3 can write the code for any AM technology.

3. CONCLUSIONS

The RP3 is a Computer-Aided Manufacturing for AM process planning that can work with the main AM categories. As it has been developed as a modular, easy-to-adapt software, it offers complete control of all heuristics to perform the process planning tasks. The proposed framework facilitates the addition of new methods for any process planning step as long as the developers adapt the defined inputs and outputs of the step to be implemented. Therefore, it is an ideal framework for research, development, and improvement in this area.

The current version of RP3 allows representing 3D models with non-topological and topological data structures, which gives the possibility of exploring methods for the slicing and support computation steps into the AM process planning. The framework is also developed so that methods can work with geometry- and image-based data structures, making it possible to adapt and propose planning strategies for different AM technologies. Filling strategies were extensively developed in the current version of RP3, which makes available basic methods such as zigzag and concentric filling and also advanced strategies to produce porous media.

There are still many improvements in this area to achieve AM's full potential and excellence, and this is the main motivation that keeps the research group working.

4. ACKNOWLEDGMENTS

This work was partially funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil (CNPq) and Financiadora de Estudos e Projetos (FINEP).

5. REFERENCES

- Berg, M.d., Cheong, O., Kreveld, M.v. and Overmars, M., 2008. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd edition.
- Chen, X. and McMains, S., 2005. "Polygon offsetting by computing winding numbers". In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 4739, pp. 565–575.
- Di Nisio, F.G., Ventura, H.T., Minetto, R., Dutra, R. and Volpato, N., 2022. "A variable bead width filling pattern to print porous media with material extrusion additive manufacturing". *The International Journal of Advanced Manufacturing Technology*, Vol. 38, p. 101760.
- Faust, R.C., Minetto, R. and Volpato, N., 2023. "Parallel tool-path generation for additive manufacturing: A gpu-based zigzag filling". *Advances in Industrial and Manufacturing Engineering*, Vol. 6, p. 100107.
- Huang, P., Wang, C.C.L. and Chen, Y., 2014. "Algorithms for Layered Manufacturing in Image Space". In *Advances in Computers and Information in Engineering Research, Volume 1*, ASME Press.

- Laskoski, J.V., Minetto, R., Volpato, N. and da Silva, R.D., 2021. “Seleção de áreas preenchíveis na estratégia tabuleiro de xadrez em impressão 3D”. In *XXVI Seminário de Iniciação Científica e Tecnológica - SICITE 2021*. UTFPR, Campus Guarapuava, Guarapuava - PR, Brazil.
- Lee, K.H. and Woo, H., 2000. “Direct integration of reverse engineering and rapid prototyping”. *Computers Industrial Engineering*, Vol. 38, No. 1, pp. 21–38.
- Machado, M.R.F.C., Volpato, N., da Silva, R.D. and Laskoski, J.V., 2023. “Análise de amostras fabricadas em aço inox 316L por L-PBF variando as estratégias de varredura”. In *12º Congresso Brasileiro de Engenharia de Fabricação - COBEF 2023*. ABCM, Brasília, DF, Brazil.
- Magalhães, L., Volpato, N. and Luersen, M., 2014. “Evaluation of stiffness and strength in fused deposition sandwich specimens”. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 36, pp. 449–459.
- Minetto, R., Volpato, N., Stolfi, J., Gregori, R.M. and da Silva, M.V., 2017. “An optimal algorithm for 3d triangle mesh slicing”. *Computer-Aided Design*, Vol. 92, pp. 1–10.
- Nakonetchnei, E.C., Stolfi, J., Volpato, N., da Silva, R.D. and Minetto, R., 2022. “Hotfill: A cooling time constrained raster-fill planning algorithm for extrusion 3d printing”. *Advances in Engineering Software*, Vol. 173, p. 103219.
- Puppi, F.M., Voltolini, R., da Silva, R.D., Volpato, N. and Minetto, R., 2022. “O uso de operações com imagens para geração de padrões 3d em manufatura aditiva”. In *XXVII Seminário de Iniciação Científica e Tecnológica da UTFPR - SICITE 2022*. UTFPR, Campus Santa Helena, Santa Helena, PR, Brazil.
- Schneider, P.J. and Eberly, D., 2002. *Geometric Tools for Computer Graphics*. Elsevier Science Inc., USA.
- Shigueoka, M.O. and Volpato, N., 2022. “Expanding manufacturing strategies to advance in porous media planning with material extrusion additive manufacturing”. *Additive Manufacturing*, Vol. 121, p. 3919–3933.
- Silvy, I.B.C., Volpato, N. and Nisio, F.G.D., 2021. “Firmware para impressora 3d por jateamento de aglutinante com dois materiais”. In *XXVI Seminário de Iniciação Científica e Tecnológica - SICITE 2021*. UTFPR, Campus Guarapuava, Guarapuava - PR, Brazil.
- Volpato, N. and Foggiatto, J.A., 2009. “The development of a generic rapid prototyping process planning system”. In *Innovative Developments in Design and Manufacturing*. Taylor Francis Group, London.
- Volpato, N., dos Santos, E.C.N., de Souza Pinto Pereira, A. and Pereira, M., 2019. “Desenvolvimento preliminar de um CAM para manufatura aditiva para o processo LMD”. In *10º Congresso Brasileiro de Engenharia de Fabricação - COBEF 2019*. ABCM - Associação Brasileira de Engenharia e Ciências Mecânicas, São Carlos - SP, Brazil.
- Volpato, N., Franzoni, A., Luvizon, D.C. and Schramm, J.M., 2013. “Identifying the directions of a set of 2d contours for additive manufacturing process planning”. *The international journal of advanced manufacturing technology*, Vol. 68, pp. 33–43.
- Volpato, N., Galvão, L.C., Nunes, L.F., Souza, R.I. and Oguido, K., 2020. “Combining heuristics for tool-path optimisation in material extrusion additive manufacturing”. *Journal of the Operational Research Society*, Vol. 71, No. 6, pp. 867–877.
- Weller, T.R., Weller, D.R., de Abreu Rodrigues, L.C. and Volpato, N., 2021. “A framework for tool-path airtime optimization in material extrusion additive manufacturing”. *Robotics and Computer-Integrated Manufacturing*, Vol. 67, p. 101999.

6. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.