Non-Uniform Planar Slicing for Robot-Based Additive Manufacturing

Jacopo Lettori[1,3] ⓘ, Roberto Raffaeli[2] ⓘ, Milton Borsato[3] ⓘ, Marcello Pellicciari[2] ⓘ and Margherita Peruzzini[1] ⓘ

[1]InterMech-DIEF, University of Modena and Reggio Emilia, jacopo.lettori@unimore.it, margherita.peruzzini@unimore.it
[2]InterMech-DISMI, University of Modena and Reggio Emilia, roberto.raffaeli@unimore.it, marcello.pellicciari@unimore.it
[3] Universidade Tecnológica Federal do Paraná, jacopo.lettori@unimore.it, borsato@utfpr.edu.br

Corresponding author: Jacopo Lettori, jacopo.lettori@unimore.it

Abstract. Planar slicing algorithms with constant layer thickness are widely implemented for geometry processing in Additive Manufacturing (AM). Since the build direction is fixed, a staircase effect is produced, decreasing the final surface finish. Also, support structures are required for overhanging portions. To overcome such limits, AM is combined with manipulators and working tables with multiple degrees of freedom. This is called Robot-Based Additive Manufacturing (RBAM) and it aims to increase the manufacturing flexibility of traditional printers, enabling the deposition of material in multiple directions. In particular, the deposition direction is changed at each layer requiring non-uniform thickness slicing. The total number of layers, as well as the volume of the support structures and the manufacturing time are reduced, while the surface finish and mechanical performance of the final product are increased. This paper presents an algorithm for non-uniform planar slicing developed in Rhinoceros and Grasshopper. It processes the input geometry and uses parameters to capture manufacturing limits. It mostly targets curved geometries to remove the need for support structures, also increasing the part quality.

Keywords: Robot-based Additive Manufacturing, Non-Uniform Slicing, Multiaxial Deposition.
DOI: https://doi.org/10.14733/cadaps.2024.104-118

1 INTRODUCTION

In Additive Manufacturing (AM) parts are usually built along a Z direction identified as the normal to an optimal slicing plane [16]. In the slicing process, a set of 2D curves that represent the tool path of the printer is generated from the input 3D model. Uniform planar slicing with constant layer thickness is widely implemented in commercial software due to its simplicity, robustness, and limited processing time [4]. However, this strategy results in the staircase effect for processed

surfaces that are tilted with respect to the slicing direction, decreasing the surface finish of the final part (see Figure 1a). The staircase effect can be evaluated by measuring the cusp height. This value describes the maximum deviation between the printed part and model surface [5].

Planar adaptive slicing algorithms were developed to control the staircase effect [19], as depicted in Figure 1b. This approach foresees the variation of the layer thickness according to the shape of the processed CAD [6], balancing the staircase effect with the manufacturing time [13], (Figure 1c). Adaptive strategies can be found in commercial slicing software. The process parameters are adapted according to the required layer thickness, considering the minimum and maximum thickness limits of the printing device.
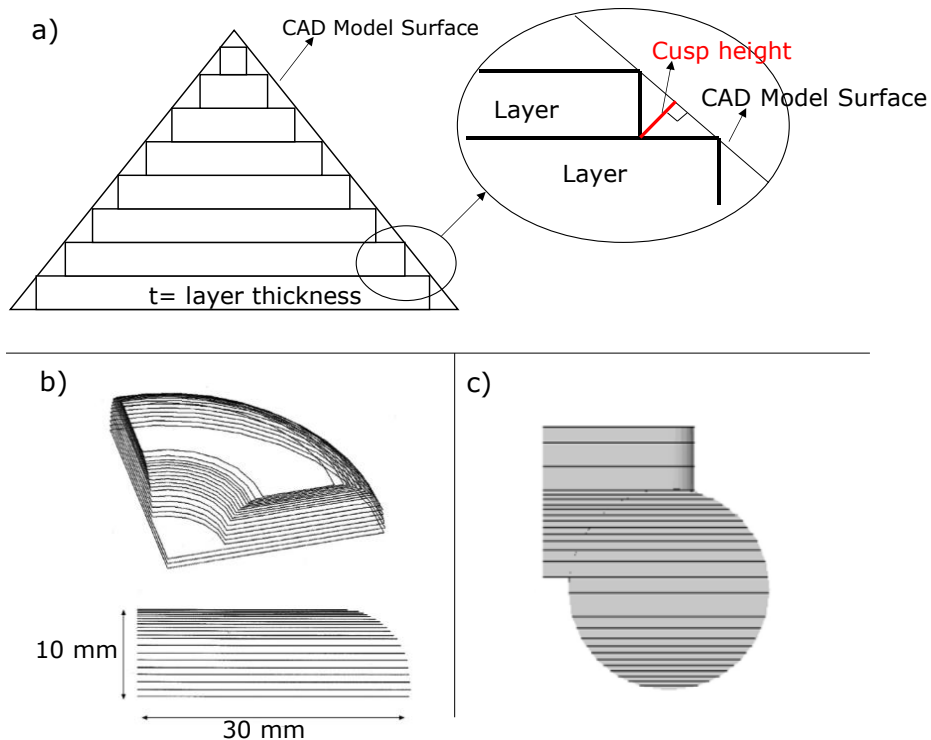
Figure 1: a) Staircase effect, taken from [10]; b) Adaptive planar slicing, taken from [19]; c) Adaptive planar slicing, taken from [13].

In addition, the adoption of a fixed slicing direction with planar uniform or adaptive slicing leads to suspended geometry regions where support structures are required [7]. Support structures must be removed by a physical, chemical, or thermal process. This post-process phase is time-consuming, reduces the surface finish, and can damage the final product.

In this context, Robot-Based Additive Manufacturing (RBAM) can overcome these limits. RBAM is the combination of an additive process and manipulators and/or working tables with multiple degrees of freedom [9]. For example, an extruder for plastic material [23] or a welding torch [17] can be attached to a robot arm. These solutions are used to increase the manufacturing flexibility of cartesian AM [10], enabling the deposition of material in multiple directions. Moreover, non-planar [3] and non-uniform thickness [25] slicing can be realized.

In particular, non-uniform slicing presents non-parallel planar layers, so the slicing direction is changed at each layer [25]. The possibility of a variable thickness in the layer extension allows much expanded manufacturing options, possibly leading to a reduction of the total number of layers, the support volume, and manufacturing time, also increasing the surface finish and

mechanical performances of the final product [8]. Even if support structures may be still required for some portions [10], their complete absence becomes feasible.

In this context, this paper presents a novel slicing strategy based on planar but non-uniform thickness layers. The approach is part of the framework presented in [11] and it can be applied after an initial volume decomposition phase [10], i.e. a subdivision of the part in regions to be processed separately. The remainder of the paper is organized as follows. The state of the art on non-uniform slicing algorithms is reported in section 2. The proposed approach is outlined in section 3. In section 4, the algorithm is applied to a simplified collector with flanges to show its robustness. Finally, conclusions and future works are presented in section 5.

2 STATE OF THE ART

Non-uniform slicing is used to process curved parts by varying the thickness within the single layer extension [25]. This approach can be implemented by changing the slicing direction for each layer to optimize some objective function (e.g., minimize the overhangs or reduce the staircase effect).

In the literature, the optimal slicing directions for the sub-volumes are identified by adopting the minimal enclosing crown algorithm, as depicted in Figure 2. An initial simplification step of small holes can be added to improve the algorithm performance [25,10]. The minimal enclosing crown algorithm is applied considering one layer at a time and collecting in a set of cardinality n a bunch of normal vectors sampled on the part surface within the layer thickness. The problem of computing the optimal slicing direction is then defined as follows: given the set of normal vectors $v = (v_1,…, v_i,…, v_n)$, find the optimal vector that produces the smallest angle between this vector and any vector v_i within the set [10]. The vectors are reported in a Gauss Map [22], so they are represented as a set of points. Practically, a spherical crown with a minimum-radius bottom surface that contains all the points is found on the sphere surface. The normal vector to the bottom surface of the minimal enclosing crown is the optimal slicing direction [2], as shown in Figure 2c.
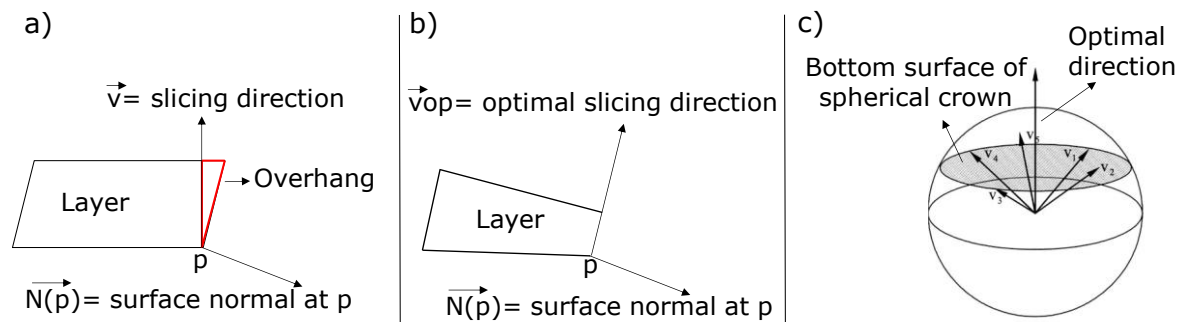
Figure 2: Minimal enclosing crown algorithm, taken from [10]: a) Result in overhang with planar uniform slicing, adapted from [25]; b) Optimal slicing direction to remove the overhang, adapted from [25]; c) Construction of the minimal enclosing crown, taken from [25].

Due to technological reasons, the layer thickness must be bounded to a specific range since deposition means are not able to exceed specific limits. Therefore, the deposition direction of a layer cannot be too tilted compared to the one of the previous one because this would lead to an excessive thickness variation within the layer.

In [25], an average slicing direction is calculated when the limits are not respected between the direction calculated through the minimal enclosing crown algorithm and the slicing direction of the previous layer. This process is iterated until the technological limits are satisfied. However, the use of an average does not guarantee that the calculated direction is the optimal one for the imposed limits. Figure 3 depicts an example of non-uniform planar slicing applied on a generic

curved geometry. Furthermore, the minimal enclosing crown algorithm can be unstable for simple geometries originating few normal directions.
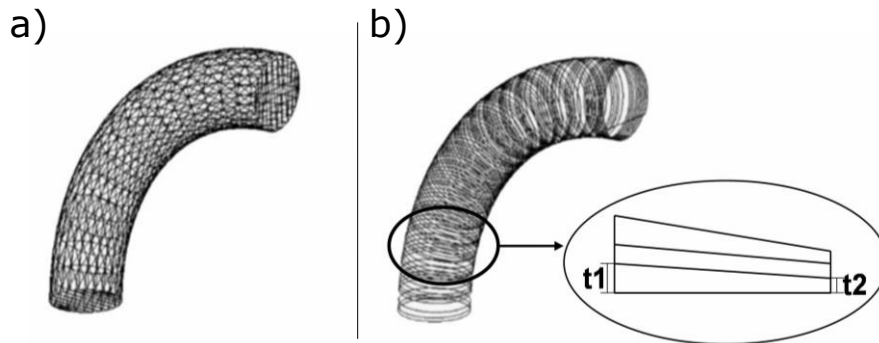
Figure 3: Non-uniform planar slicing of a curved geometry, taken from [10]: a) Input geometry, taken from [25]; b) Thickness of the layer ranging from t1 to t2, taken from [25].

Alternatively, non-uniform slicing can be driven by the centroid axis [18] or 1D medial axis [20] of the input geometry. The centroid axis is composed of a series of points that are centroids of cross sections at different locations and whose positions can be obtained by computing the barycenter of a cross-section (Figure 4a). In [10] when overhangs areas are identified, the centroid of the lower cross section is connected to all the centroids of the overhang area portions, creating a certain number of vectors *V*. Since the normal direction is restricted to X, Y or Z axis, the new normal direction of the intersecting plane is one of the candidate directions which forms a minimum angle with the group of vectors *V*. The centroid can be used as a dorsal curve for slicing, defining curved trajectories. However, extremal or bulky portions are often not properly captured, as shown in Figure 4b.

On the other hand, the medial axis algorithm can be used to drive a continuous multiaxial deposition [1]. The medial axis, also called skeleton 1D [20], is defined as the locus of points that correspond with the centers of the maximum diameter spheres enclosed in the geometry [15]. For instance, the skeleton can be obtained by mesh contraction [1]. Similarly to the centroid axis algorithm, the medial axis can be used to identify the slicing direction for non-uniform planar slicing [21] (Figure 4c) or non-uniform and non-planar slicing [24]. However, this strategy is mainly oriented to tree-like structures. Also, the medial axis identification depends on the quality of the mesh and could be computationally expensive to be obtained [1].

Non-uniform thickness is also implemented as non-planar layers to increase the surface finish and the adhesion among layers [3,12,14]. However, the present work is limited to non-uniform planar layers.

3   PROPOSED ALGORITHM

A slicing algorithm is presented here to process geometries according to variable deposition direction and restricted thickness interval due to manufacturing limits. The inputs of the algorithm are the initial geometry in B-Rep format, a reference layer height (*hLayer*), and the maximum and minimum thickness that can be obtained with the selected technology (*hMin*, *hMax*). The overall algorithm is reported in Figure 5 and as pseudocode in Algorithm 1. A generic index *i* as a subscript of a variable indicates the *i-th* element of an ordered set. Functions present in Grasshopper and other related free plug-ins, as well as routines created based on these functions have been used.
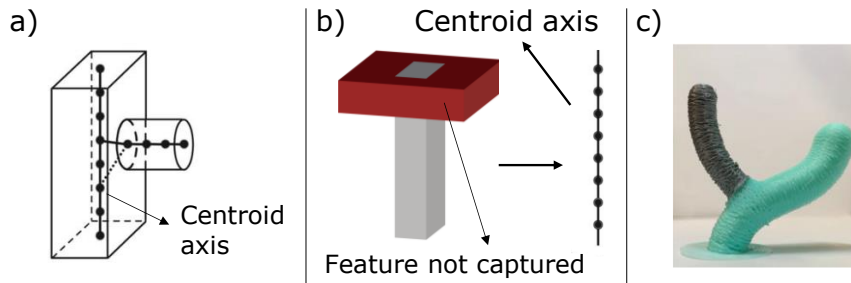
Figure 4: Non-uniform slicing driven by medial axis: a) Centroid axis extraction, taken from [18]; b) Issue with centroid axis algorithm, taken from [18]; c) Result of deposition process [21].
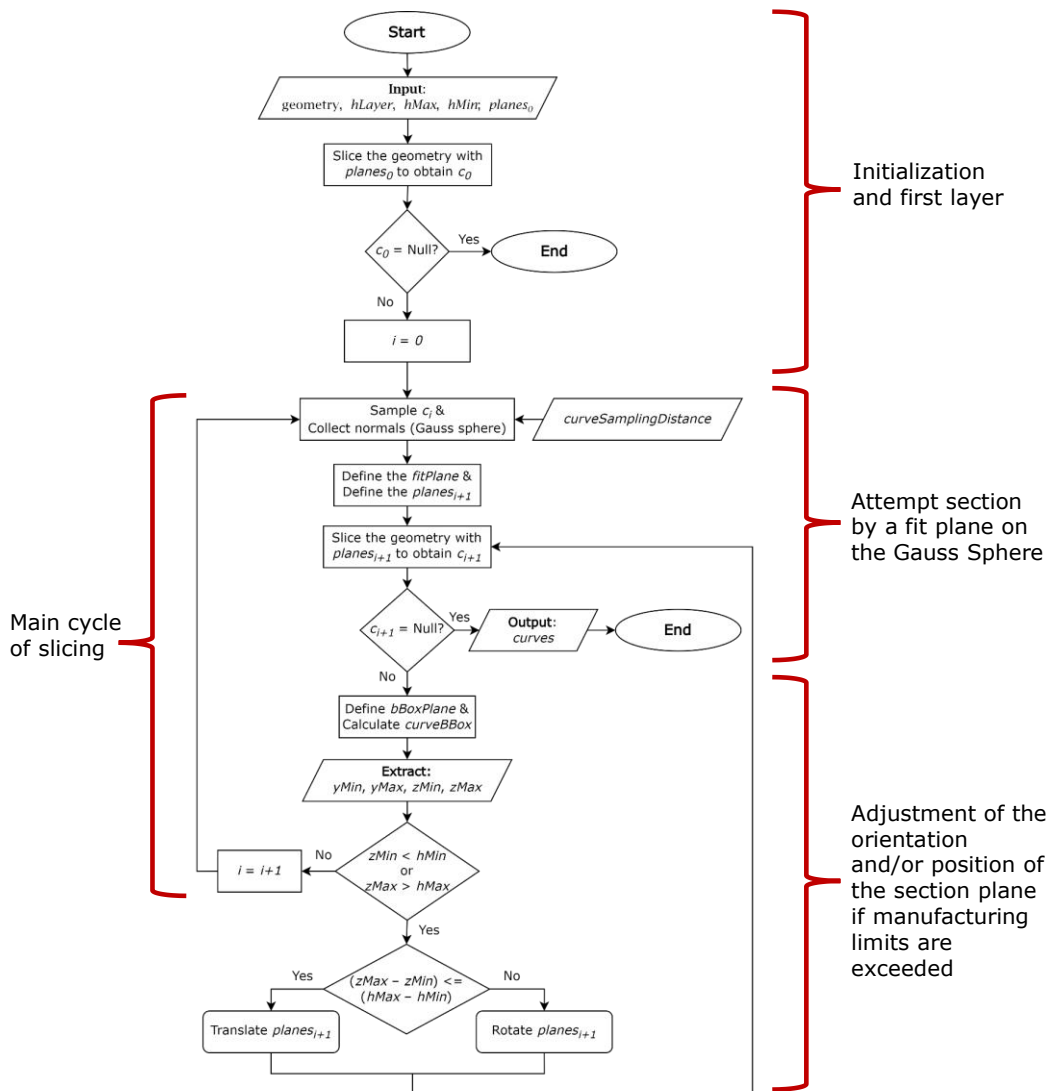
Figure 5: Overall flowchart of the proposed algorithm.

ALGORITHM 1: NON-UNIFORM PLANAR SLICING

in:	$brep$, $hLayer$, $hMin$, $hMax$, initial deposition plane
out:	$c[]$, set of the outer borders of the layers
local:	$curveSamplingDistance \leftarrow$ choose a sampling distance proportional to $hLayer$
	$planes[] \leftarrow$ set of section planes
	$i \leftarrow 0$ $\triangleright$ Layer counter

$\triangleright$ Initial validity checks
1: **if** ($hLayer < hMin$) $\vee$ ($hLayer > hMax$) **then**
2:    **error** wrong parameters
3: **end if**
4: $planes_0 \leftarrow$ initial deposition plane chosen by the user

$\triangleright$ Add first section curve
4: $c_0 \leftarrow$ INTERSECTION($brep$, $planes_0$)
5: ADDITEM(c, c_0)

$\triangleright$ Main loop to compiute the layers sequence
6: **while** $c_i \neq$ NIL **do**
7:    $n_i \leftarrow$ GETPLANENORMAL($planes_i$)
8:    $points \leftarrow$ DIVIDEBYDISTANCE(c_i, $curveSamplingDistance$)
9:    $normals \leftarrow$ GETNORMALS($points$, $brep$)
10:    $fitPlane \leftarrow$ FITPLANEONGAUSSSPHERE($normals$)
11:    $d_i \leftarrow$ GETPLANENORMAL($fitPlane$)
12:    $O_i \leftarrow$ GETPLANEORIGIN($fitPlane$)
13:    $O_{i+1} \leftarrow O_i + d_i * hLayer$
14:    $planes_{i+1} \leftarrow$ CREATEPLANEBYNORMAL(O_{i+1}, d_i)
15:    $c_{i+1} \leftarrow$ INTERSECTION($brep$, $planes_{i+1}$)
16:    **if** $c_{i+1} =$ NIL **then**
17:       **return** c $\triangleright$ The part processing is finished
18:    **end if**

$\triangleright$ Evaluate curve orientation
19:    $xAxis \leftarrow$ CROSSPRODUCT(d_i, n_i)
20:    $yAxis \leftarrow$ CROSSPRODUCT($xAxis$, n_i)
21:    $bBoxPlane \leftarrow$ CREATEPLANE (O_i, $xAxis$, $yAxis$)
22:    $curveBBox \leftarrow$ GETBOUNDINGBOX(c_{i+1}, $bBoxPlane$)
23:    $<yMin, yMax, zMin, zMax> \leftarrow$ BOUNDINGBOXEXTENSION($curveBBox$)

$\triangleright$ Iterate while $plane_{i+1}$ position respects manufacturing limits
24:    **while** ($zMin < hMin$) $\vee$ ($zMax > hMax$) **do**

                 $\triangleright$ The plane needs to be translated only
25:         **if** ($zMax - zMin$) $<=$ ($hMax - hMin$) **then**
26:            **if** $zMin < hMin$ **then**
27:               $planes_{i+1} \leftarrow$ TRANSLATEPLANE($planes_{i+1}$, ($hMin - zMin$) $* n_i$)
28:            **else**
29:               $planes_{i+1} \leftarrow$ TRANSLATEPLANE($planes_{i+1}$, ($hMax - zMax$) $* n_i$)
30:            **end if**

```
31:                          c_{i+1} ← INTERSECTION(brep, planes_{i+1})
32:                          if c_{i+1} = NIL then
33:                                  return c                    ▷ The part processing is finished
34:                          end if
35:                          curveBBox ← GETBOUNDINGBOX(c_{i+1}, bBoxPlane)
36:                          <yMin, yMax, zMin, zMax> ← BOUNDINGBOXEXTENSION(curveBBox)

                    ▷ The plane needs to be rotated
37:                  else
38:                          if (zMax > hMax) ∨ (zMin < hMin) then
39:                                  angle ← ARCTAN(((zMax - hMax) – (zMin - hMin)) / (yMax - yMin))
40:                                  d_i ← ROTATE(d_i, xAxis, angle)
41:                                  planes_{i+1} ← CREATEPLANEBYNORMAL(O_{i+1}, d_i)
42:                                  c_{i+1} ← INTERSECTION(brep, planes_{i+1})
43:                                  if c_{i+1} = NIL then
44:                                          return c                    ▷ The part processing is finished
45:                                  end if
46:                                  xAxis ← CROSSPRODUCT(d_i, n_i)        ▷ Section plane rotation axis
47:                                  yAxis ← CROSSPRODUCT(xAxis, n_i)
48:                                  bBoxPlane ← CREATEPLANE (O_i, xAxis, yAxis)
49:                                  curveBBox ← GETBOUNDINGBOX(c_{i+1}, bBoxPlane)
50:                                  <yMin, yMax, zMin, zMax> ← BOUNDINGBOXEXTENSION(curveBBox)
51:                          end if
52:                  end if
53:          end while
54:          if c_{i+1} = NIL then
55:                  return c                                    ▷ The part processing is finished
56:          end if

                    ▷ The curve can be added to the output set
57:          ADDITEM(c, c_{i+1})
59:          i ← i +1
60:  end while
```

For simplicity, the algorithm will be described referring to the first two layers, whose geometrical entities are referred to with indices 0 and 1, respectively. The algorithm provides a set of curves *c* as output, which corresponds to the outer bounds of each layer to be deposited. The actual deposition path is computed in a successive phase, which is not in the scope of this paper, according to the required infill strategy.

After placing the part on an initial support plane ($planes_0$) with normal n_0 which is defined by the user, the slicing algorithm begins by intersecting it with the geometry, obtaining the first curve c_0 which corresponds to the external path of the first layer (Figure 6).

c_0 is sampled along its length at a fixed distance (constant value), namely *curveSamplingDistance*, to obtain a set of points. This distance is chosen so that it is proportional to *hLayer*. Normal vectors to the processed geometry are computed for each point, as depicted in Figure 7a. The vectors are collected in a unit Gaussian Sphere, so that each vector is represented as a point on the sphere (Figure 7b). The plane that best fits (*fitPlane*) this set of points is calculated. The normal to *fitPlane* identifies the new slicing direction d_0. The origin of the plane O_0 is moved along the d_0 by a quantity equal to the imposed layer height *hLayer*, creating a new point O_1. Then, the plane $planes_1$ with origin O_1 and normal d_0 is created, as shown in Figure 7c. The next curve c_1 is obtained by the intersection of the initial geometry with $planes_1$.
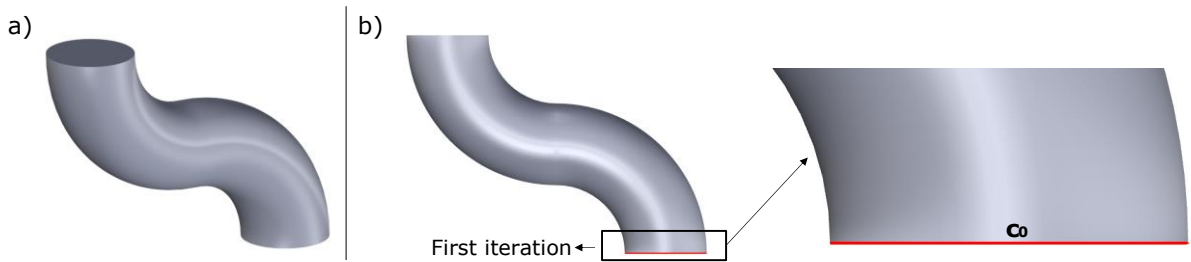
Figure 6: Initial step of the slicing algorithm: a) Input geometry which is conveniently oriented; b) First slicing curve from an initial section plane.

After, the plane *bBoxPlane* is defined which is oriented so that it lays on *planes$_0$,* but its x axis is aligned with the rotation axis between *planes$_0$* and *planes$_1$*, i.e., *xAxis* (Figure 7d). The bounding box *curveBBox* of c_1 aligned to the plane *bBoxPlane* is then computed, and the maximum and minimum values of its corners are considered (respectively *yMin*, *yMax*, *zMin* and *zMax*). The extremes of the Z domain correspond to the maximum and minimum thickness required to manufacture the layer. When the maximum and minimum values exceed the imposed limits, a translation and/or a rotation are applied to the *planes$_1$*.
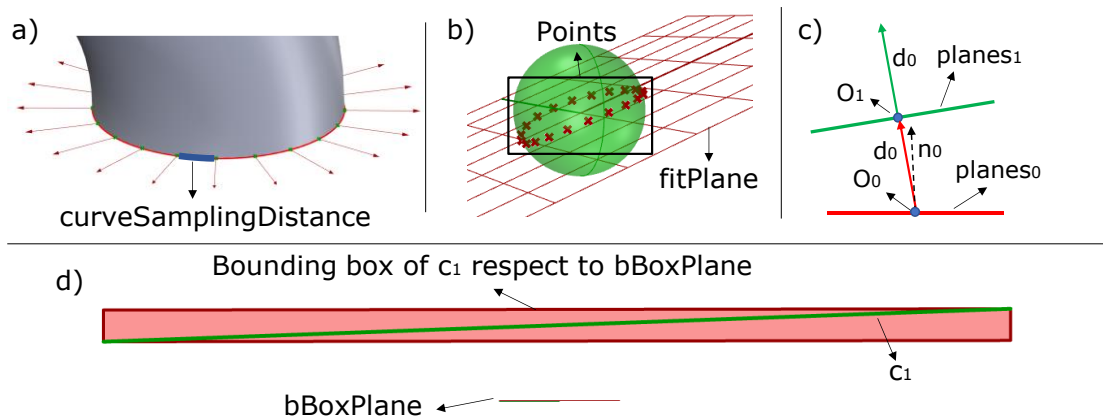
Figure 7: Description of the non-uniform slicing algorithm: a) Surface normal vectors of the input geometry at curve c_0; b) Collection of surface normal vector in a unit sphere and fitting of a plane; c) Creation of the *planes$_1$*; d) Evaluation of the axes aligned bounding box of c_1 to verify the required thickness of the beads.

In particular, a translation is applied when the difference between *zMax* and *zMin* is lower than the difference between the imposed limits *hMin* and *hMax*. This is because a simple translation may be enough to bring the required bead thickness to the limits. If *zMax* is greater than *hMax*, the *planes$_1$* is translated by a distance equal to the difference between *hMax* and *zMax*. If *zMin* is lower than *hMin*, the translation distance is equal to the difference between *hMin* and *zMin*. On the other hand, a rotation is required when the difference between *zMax* and *zMin* is greater than the difference between *hMax* and *hMin*. In this case, the d_0 is rotated along the vector which is perpendicular to both n_0 and d_0. A new *planes$_1$* is calculated with origin equal to O_1 and normal equal to the rotated d_0.

In both cases, a new bounding box is calculated, and its Y and Z domains are analyzed again to be sure that the new sectioning position leads to acceptable manufacturing limits. The algorithm

converges after a few iterations, guaranteeing a fast identification of an optimal slicing plane. So, the processing time is strongly reduced compared to the algorithm presented in [25]. Once $planes_1$ is determined, it is intersected with the initial geometry to obtain c_1, defining a non-uniform layer as in Figure 8a. Finally, the algorithm identifies the following $planes_i$ and c_i, repeating the procedure until all the geometry is sliced. A typical result is presented in Figure 8b.
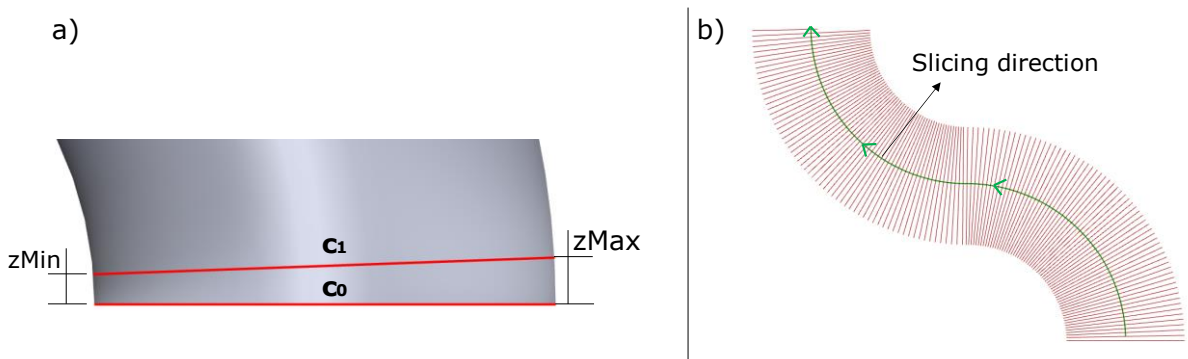
Figure 8: a) Single non-uniform layer; b) Typical result of the slicing process on the selected curved geometry.

Note that the algorithm can process curvatures that do not necessarily lies in only one plane as reported in the example reported in Figure 9. However, as a limitation of the algorithm, it is worth noting that more than one curve can be obtained when intersecting the input solid with a plane. It consists of inner loops due to cavities, and outer curves of other portions of the solid. At the moment, the algorithm assumes the presence of only one external curve. However, it can easily be extended to handle multiple section curves.
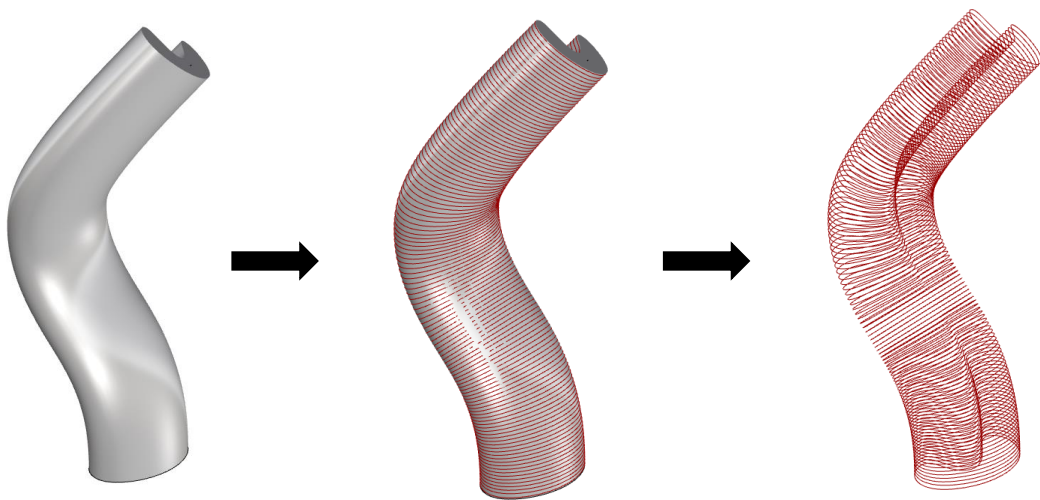

Figure 9: Non-planar slicing applied to a swept geometry whose dorsal curve presents curvatures in multiple planes.

The algorithm has been implemented in the Rhinoceros version 7 CAD environment taking advantage of the Grasshopper plug-in for visual programming. In particular, the GHPython

interpreter was used to develop the slicing routine, integrating Python scripts with the native functions of the Grasshopper library.

The developed Grasshopper block is called "Non-Uniform Slicing" and is represented in Figure 10. The block presents four inputs and one output. The inputs are the geometry to be processed (*brep*), the layer height (*hLayer*), and the maximum (*hMax*) and minimum (*hMin*) heights that can be obtained with a certain technology. All these parameters can be changed according to the selected technology and knowledge of the user. The output is a set of slicing curves.
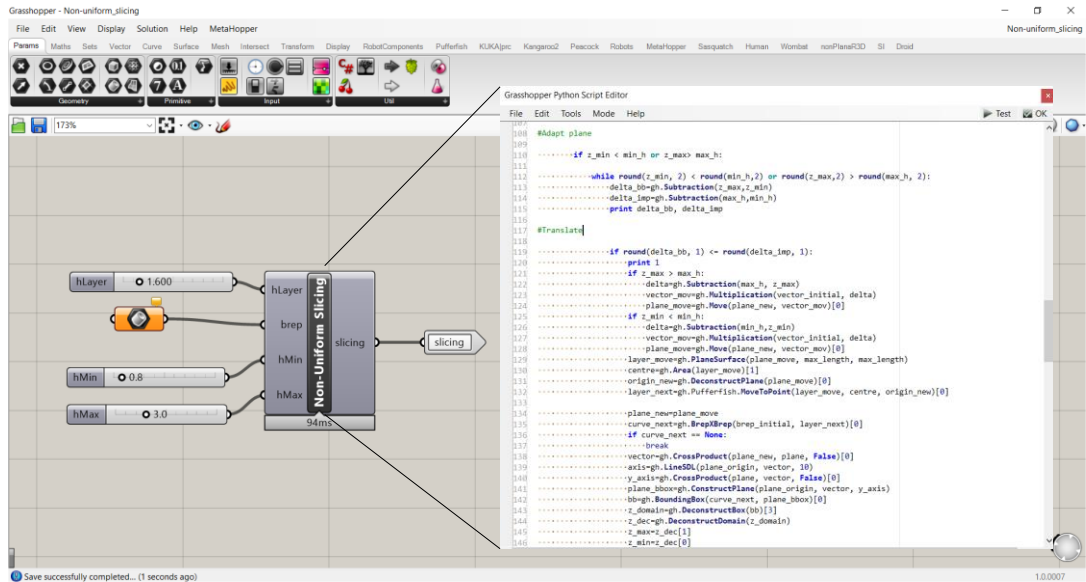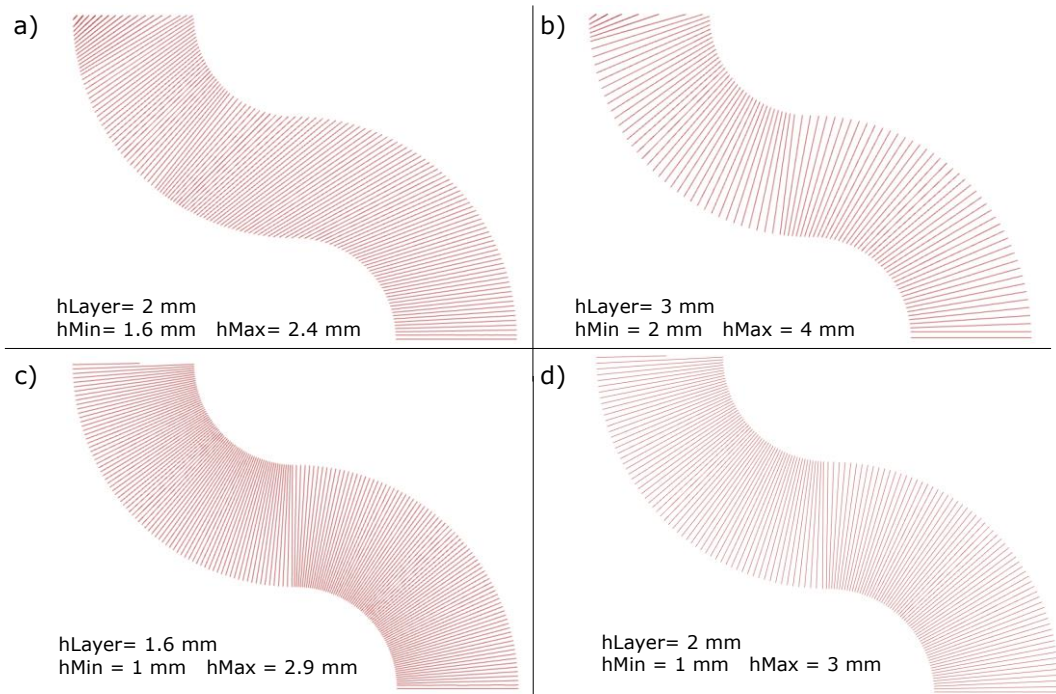

Figure 10: Developed algorithm in the plug-in of Grasshopper by using a GHPython block.

The plug-in allowed to test various parts to validate the approach and evaluate the influences of the layer thickness limits to the slicing result, as shown in Figure 11. Note that the algorithm provides a uniform planar slicing when the three required parameters for the technological limits are equal to the *hLayer*. Also, if *hMin* is greater than *hLayer* or *hMax* is lower than *hLayer* the algorithm provides an error message.

4   TESTING OF THE ALGORITHM

Two geometries have been selected to evaluate the reliability of the algorithm. The first test case is a pipe that presents curvatures in multiple planes where the holes are simplified, as depicted in Figure 12a. The bounding box of the part measures 215x133x174 mm. A constant thickness of 2 mm was used in the uniform planar slicing and 87 layers were found. However, overhangs are encountered, and most of the external surface is subjected to a significant staircase effect (Figure 12b).

The slicing result of the proposed algorithm is presented in Figure 13. The algorithm adapts the slicing planes according to the imposed technological limits, increasing the performance of the algorithm. The limits have been chosen in the range between 1 and 3 mm while the layer height is set as 2 mm. Indeed, the slicing direction can follow the multi-plane curvature of the geometry. Despite the total number of layers is higher (147 layers) than the planar uniform slicing, the proposed algorithm removes the need of support, and it highly increases the surface finish and geometry accuracy. So, the proposed approach is preferable to the planar uniform slicing strategy.

a)
hLayer= 2 mm
hMin= 1.6 mm   hMax = 2.4 mm

b)
hLayer= 3 mm
hMin = 2 mm   hMax = 4 mm

c)
hLayer= 1.6 mm
hMin = 1 mm   hMax = 2.9 mm

d)
hLayer= 2 mm
hMin = 1 mm   hMax = 3 mm

Figure 11: Result of the non-uniform slicing applied to the same geometry but with different manufacturing limits.

a)

b)
Staircase effect
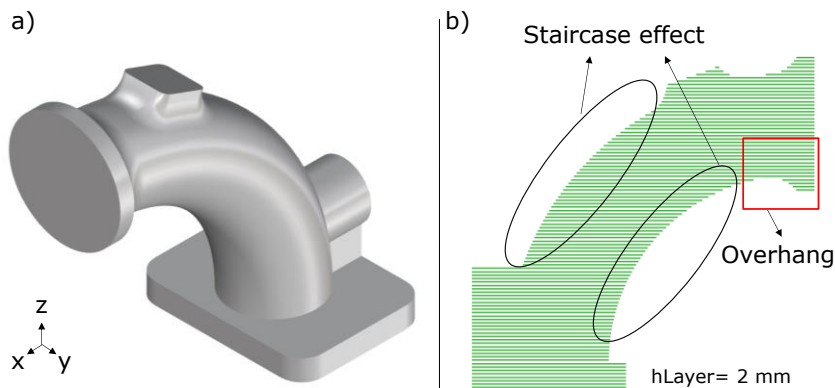
Overhang

hLayer= 2 mm
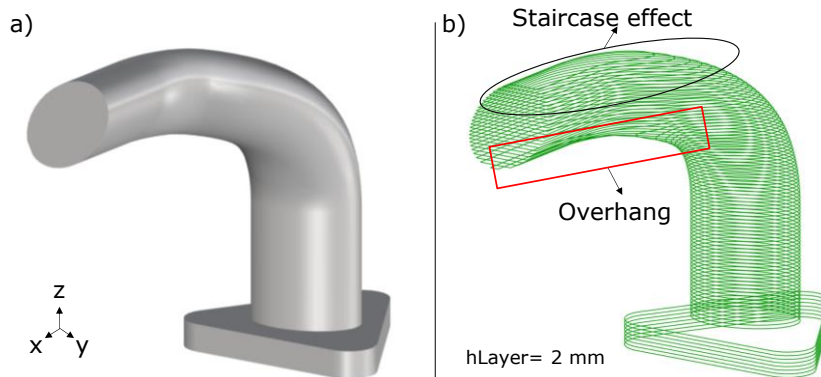
Figure 12: Test case 1: a) Simplified bent pipe; b) Result by applying planar uniform slicing with constant thickness.

The second test case resembles a collector with flanges where internal cavities have been removed (see Figure 14a). The bounding box measures are 200x120x195 mm. By implementing a uniform planar slicing with a constant thickness of 2 mm a total number of 106 layers is found. Also in this case, overhangs are encountered, and most of the external surface is subjected to a significant staircase effect (Figure 14b).

✓ **No overhangs**

✓ **Reduction of the staircase effect**

hLayer= 2 mm   hMin= 1 mm   hMax= 3 mm
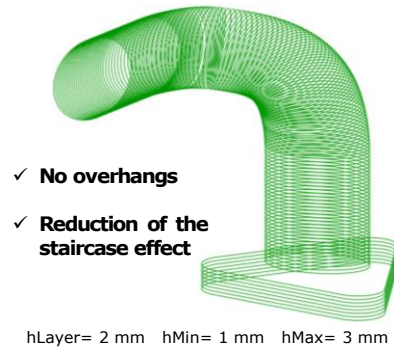
Figure 13: Test case 1: sequence of slicing curves resulting from the application of the proposed approach.

Figure 14: Test case 2: a) Simplified collector with flanges; b) Result by applying planar uniform slicing with constant thickness.

By applying the proposed algorithm, overhangs are completely removed, and the staircase effect is considerably reduced (see Figure 15). Also in this example, the algorithm adapts the slicing planes according to the imposed technological limits, which have been chosen in the range between 1.4 and 3 mm. Using such parameters, the iterations to find section planes are strongly reduced, since a maximum of 2 iterations is enough for each layer.

The resulting total number of layers is 130, which is slightly higher than the uniform planar slicing approach. That means a higher manufacturing time. However, support structures are not required in non-uniform planar slicing. Support structures lead to a waste of materials, also increasing the printing time and post processing time, so the proposed solution is certainly more efficient.

The computational time results are proportionally linked to the layer height, the technological limits, and the dimensions of the CAD model. For the proposed parts the processing time is below 2 minutes, but it is strongly affected by the adopted technology for the algorithm prototype. A standard implementation using C++ coding language would allow much more time savings.

5   CONCLUSIONS AND FUTURE WORKS

Non-uniform slicing is a powerful approach in RBAM to process geometries, especially curved ones. It is beneficial to increase the adhesion of layers, the surface finishing, and the mechanical properties of the manufactured part, also reducing the need for support structures. This paper has introduced an algorithm for non-uniform planar slicing which leverages the RBAM possibility of continuously varying the deposition direction and the thickness of the deposited material bead

within each single layer. Four inputs are required to perform non-uniform slicing: layer height, the initial geometry, and the maximum and minimum layer thickness that can be manufactured with the selected technology. The output of the algorithm is a set of non-uniform slicing curves. The algorithm is based on the surface orientation of the input CAD geometry. Unlike other approaches, resulting layers minimize the cusp heights at the borders. Also, the algorithm adapts the slicing plane according to technological limits, decreasing the overall runtime of the algorithm. Also, it does not require the extraction of the part medial axis.

✓ **No overhangs**

✓ **Reduction of the staircase effect**

hLayer= 2 mm   hMin= 1.4 mm   hMax= 3 mm
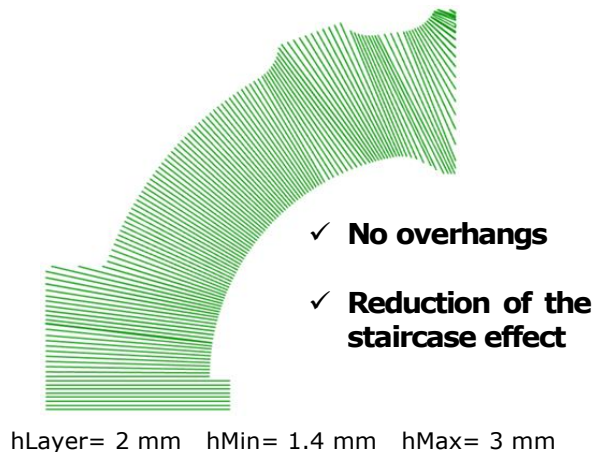
Figure 15: Test case 2: sequence of slicing curves resulting from the application of the proposed approach.

The algorithm was tested in two simplified geometries, to show its ability in following the shape of the part minimizing the lateral surface quality. In future works, specific hardware solutions (i.e., extruder/welding torch attached to a manipulator) will be implemented to proceed with physical parts realization. Also, an extensive experimental campaign is mandatory to connect the process parameters to the desired layer height. In fact, it is necessary to continuously adjust the process parameters to obtain different layer heights, allowing for non-uniform slice thicknesses. Although, optimal infill strategies for each layer must be explored.

Jacopo Lettori, https://orcid.org/0000-0002-8523-8469
Roberto Raffaeli, https://orcid.org/0000-0003-0301-454X
Milton Borsato, https://orcid.org/0000-0002-3607-8315
Marcello Pellicciari, https://orcid.org/0000-0003-2578-4123
Margherita Peruzzini, https://orcid.org/0000-0003-2260-0392

REFERENCES

[1]   Au, O.K.C.; Tai, C.L.; Chu, H.K.; Cohen-Or, D.; Lee, T.Y.: Skeleton extraction by mesh contraction, ACM Transactions on Graphics, 27, 2008, 1–10. https://doi.org/10.1145/1360612.1360643.
[2]   De Berg, M.; Cheong, O.; Van Kreveld, M.; Overmars, M.: Computational geometry: Algorithms and applications, Springer Science & Business Media, 2008,. https://doi.org/10.1007/978-3-540-77974-2.
[3]   Chen, L.; Chung, M.F.; Tian, Y.; Joneja, A.; Tang, K.: Variable-depth curved layer fused deposition modeling of thin-shells, Robotics and Computer-Integrated Manufacturing, 57, 2019, 422–434. https://doi.org/10.1016/j.rcim.2018.12.016.

[4] Choi, S.H.; Kwok, K.T.: A tolerant slicing algorithm for layered manufacturing, Rapid Prototyping Journal, 8, 2002, 161–179. https://doi.org/10.1108/13552540210430997.

[5] Dolenc, A.; Mäkelä, I.: Slicing procedures for layered manufacturing techniques, Computer-Aided Design, 26, 1994, 119–126. https://doi.org/10.1016/0010-4485(94)90032-9.

[6] Gohari, H.; Kishawy, H.; Barari, A.: Adaptive variable layer thickness and perimetral offset planning for layer-based additive manufacturing processes, International Journal of Computer Integrated Manufacturing, 34, 2021, 964–974. https://doi.org/10.1080/0951192X.2021.1946854.

[7] Huang, B.; Singamneni, S.: Curved layer fused deposition modeling with varying raster orientations, Applied Mechanics and Materials, 446–447, 2014, 263–269. https://doi.org/10.4028/www.scientific.net/AMM.446-447.263.

[8] Huang, B.; Singamneni, S.B.: Curved layer adaptive slicing (CLAS) for fused deposition modelling, Rapid Prototyping Journal, 21, 2015, 354–367. https://doi.org/10.1108/RPJ-06-2013-0059.

[9] Jiang, J.; Newman, S.T.; Zhong, R.Y.: A review of multiple degrees of freedom for additive manufacturing machines, International Journal of Computer Integrated Manufacturing, 34, 2021, 195–211. https://doi.org/10.1080/0951192X.2020.1858510.

[10] Lettori, J.; Raffaeli, R.; Bilancia, P.; Peruzzini, M.; Pellicciari, M.: A review of geometry representation and processing methods for cartesian and multiaxial robot-based additive manufacturing, The International Journal of Advanced Manufacturing Technology, 123, 2022, 3767–3794. https://doi.org/10.1007/s00170-022-10432-8.

[11] Lettori, J.; Raffaeli, R.; Peruzzini, M.; Pellicciari, M.: A Framework for Hybrid Manufacturing in Robotic Cells, Computer-Aided Design & Applications, 19, 2022, 1029–1041. https://doi.org/10.14733/cadaps.2022.1029-1041.

[12] Li, Y.; Tang, K.; He, D.; Wang, X.: Multi-Axis Support-Free Printing of Freeform Parts with Lattice Infill Structures, Computer-Aided Design, 133, 2021, 102986. https://doi.org/10.1016/j.cad.2020.102986.

[13] Mao, H.; Kwok, T.H.; Chen, Y.; Wang, C.C.L.: Adaptive slicing based on efficient profile analysis, Computer Aided Design, 107, 2019, 89–101. https://doi.org/10.1016/j.cad.2018.09.006.

[14] Pelzer, L.; Hopmann, C.: Additive manufacturing of non-planar layers with variable layer height, Additive Manufacturing, 37, 2021, 101697. https://doi.org/10.1016/j.addma.2020.101697.

[15] Ramanathan, M.; Gurumoorthy, B.: Interior Medial Axis Transform computation of 3D objects bound by free-form surfaces, Computer Aided Design, 42, 2010, 1217–1231. https://doi.org/10.1016/j.cad.2010.08.006.

[16] Ransikarbum, K.; Pitakaso, R.; Kim, N.; Ma, J.: Multicriteria decision analysis framework for part orientation analysis in additive manufacturing, Journal of Computational Design and Engineering, 8, 2021, 1141–1157. https://doi.org/10.1093/jcde/qwab037.

[17] Rauch, M.; Hascoet, J.-Y.; Querard, V.: A Multiaxis Tool Path Generation Approach for Thin Wall Structures Made with WAAM, Journal of Manufacturing and Materials Processing, 5(4), 2021, 128. https://doi.org/10.3390/jmmp5040128.

[18] Ruan, J.; Sparks, T.E.; Panackal, A.; Liou, F.W.; Eiamsa-Ard, K.; Slattery, K.; Chou, H.N.; Kinsella, M.: Automated slicing for a multiaxis metal deposition system, Journal of Manufacturing Science and Engineering, Transactions of the ASME, 129, 2007, 303–310. https://doi.org/10.1115/1.2673492.

[19] Sabourin, E.; And, S.A.H.; Bøhn, J.H.: Adaptive slicing using stepwise uniform refinement, Rapid Prototyping Journal, 2, 1996, 20–26. https://doi.org/https://doi.org/10.1108/13552549610153370.

[20] Tagliasacchi, A.; Delame, T.; Spagnuolo, M.; Amenta, N.; Tagliasacchi, A.; Delame, T.; Spagnuolo, M.; Amenta, N.; Telea, A.: 3D Skeletons : A State-of-the-Art Report, Computer Graphics Forum, 35, 2016, 573–597. https://hal.archives-ouvertes.fr/hal-01300281/file/3D_Skeletons_STAR.pdf.

[21] Wang, X.; Chen, L.; Lau, T.-Y.; Tang, K.: A skeleton-based process planing framework for

support-free 3+2-axis printing of multi-branch freeform parts, The International Journal of Advanced Manufacturing Technology, 110, 2020, 327–350. https://doi.org/10.1007/s00170-020-05790-0.

[22] Woo, T.C.: Visibility maps and spherical algorithms, Computer-Aided Design, 26, 1994, 6–16. https://doi.org/10.1016/0010-4485(94)90003-5.

[23] Wu, C.; Dai, C.; Fang, G.; Liu, Y.J.; Wang, C.C.L.: General Support-Effective Decomposition for Multi-Directional 3-D Printing, IEEE Transactions on Automation Science and Engineering, 17, 2020, 599–610. https://doi.org/10.1109/TASE.2019.2938219.

[24] Xie, F.; Jing, X.; Zhang, C.; Chen, S.; Bi, D.; Li, Z.; He, D.; Tang, K.: Volume decomposition for multi-axis support-free and gouging-free printing based on ellipsoidal slicing, Computer-Aided Design, 143, 2022, 103135. https://doi.org/10.1016/j.cad.2021.103135.

[25] Zhang, J.; Liou, F.: Adaptive slicing for a multi-axis laser aided manufacturing process, Journal of Mechanical Design, Transactions of the ASME, 126, 2004, 254–261. https://doi.org/10.1115/1.1649966.