L. Robles-Lorite, R. Dorado-Vicente, A. García-Collado, E. Torres-Jiménez, Procedure to improve the surface fidelity of additive manufactured parts via direct slicing tessellation, Expert Systems with Applications, Volume 242, 2024,122822, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.122822

Procedure to Improve the Surface Fidelity of Additive Manufactured Parts via Direct Slicing Tessellation

L. Robles-Lorite[1], R. Dorado-Vicente[2], A. García-Collado[3], E. Torres-Jiménez[4]

Department of Mechanical and Mining Engineering, University of Jaén

EPS de Jaén, Campus Las Lagunillas, 23071, Jaén, Spain

[1]Laura.rlorite@gmail.com; [2]rdorado@ujaen.es; [3]acollado@ujaen.es; [4] etorres@ujaen.es

Abstract

Part appearance issues due to dimensional and shape inaccuracies (fidelity) hinder using additive manufacturing processes to obtain final products. Inaccuracies come from process parameter definition, post-processing operations, and the slicing process. Regarding the slicing procedure, the conventional approach is the indirect slicing of a tessellation that approximates the part model. Layer computation is straightforward in this method but introduces dimensional deviations for curvilinear surfaces. Direct slicing minimizes these deviations, but computation and layer representation are complex. This work blends both solutions explaining how to build an improved tessellation from direct slices that can be introduced and processed in usual slicing programs. Two shapes were printed and measured using a 3D scanner. The deviation between the CAD model and the representation on the resulting surfaces of zebra stripes and mean curvature show that the proposed solution enhances the printed surface fidelity compared to the conventional indirect slicing.

1. Introduction

Nowadays, Additive Manufacturing (AM) processes provide products manufactured by adding material in layers [1]. The AM approach requires a few tools and operations, which help to obtain products with complex geometries [2]. These characteristics reduce small series' development and production time compared to conventional manufacturing processes. Additionally, AM is specially indicated to personalize the product, design, and optimize it

regarding the assembly, since it reduces the number of parts required. Among the existing AM processes, material extrusion is probably the most widespread, especially the Fused Filament Fabrication (FFF) technology, which is being implemented in a wide range of applications, from tool manufacturing [3] to aerospace structures [4].

Among other disadvantages, dimensional accuracy limitations [5], the stair effect [6], and mechanical anisotropy [7] make it difficult to use AM processes to obtain final products. These drawbacks are in the spotlight of the research community because there are no closed rules yet to get optimal results. Although each process stage of AM (pre-process, process, and post-process) contributes to the final disagreement between theoretical and printed products, most studies focus on pre-processing. These works tackle the manufacturing process parameters to improve not only the mechanical properties but also the resulting products, dimensions, and roughness [8]. In addition, several authors have focused on detecting the source of errors [9, 10] and improving them using different techniques, especially machine learning [11, 12].

The slicing strategy required in the pre-processing stage is a less studied source of dimensional and form errors [13] concerning all AM processes. AM procedures start with a CAD model of a product. Next, a CAD plugin or a specific software slices the model and, for each specific AM technology, builds a program of instructions using G-codes to control the tool (path, speed, and acceleration) and the needed process parameters [14].

The slicing is usually accomplished on a discretized approximation of the CAD model: indirect slicing. The discretization in the STL format, likely the standard representation in AM [15, 16], is fast and fully integrated into most CAD programs. While this format perfectly fits prismatic products, it is inefficient for parts defined by curvilinear surfaces, which are present in many practical applications [17]. Curvilinear surfaces require fine meshes to limit the deviation between the CAD model and the discretization. CAD programs control the discretization with different parameters and use different default values [18, 19]. In most software, the STL file export settings depend on the chordal tolerance, the angle control, and the type of format (ASCII or binary) [20].

The indirect slicing procedure simplifies the intersection computations and the infill pattern generation. Nevertheless, due to the discrete nature of STL, this method is the source of two types of dimensional errors already mentioned by Oropallo and Piegl [21]. Consider the

example in Fig. 1; it is possible to note two types of dimensional errors produced by the indirect slicing of curvilinear surfaces:

- Error type *a*. Deviations in the slicing (cutting) direction noted when polygonal' vertices found in the intersection with the STL approximation are on the model intersection curve.
- Error type *b*. These deviations arise when polygonal' vertices found in the intersection with the STL approximation are not on the model intersection curve. This error comes from the differences between the model and its discretization in the normal direction to the slicing plane.

The above errors can also produce shape drawbacks because of the deviations from the theoretical model.

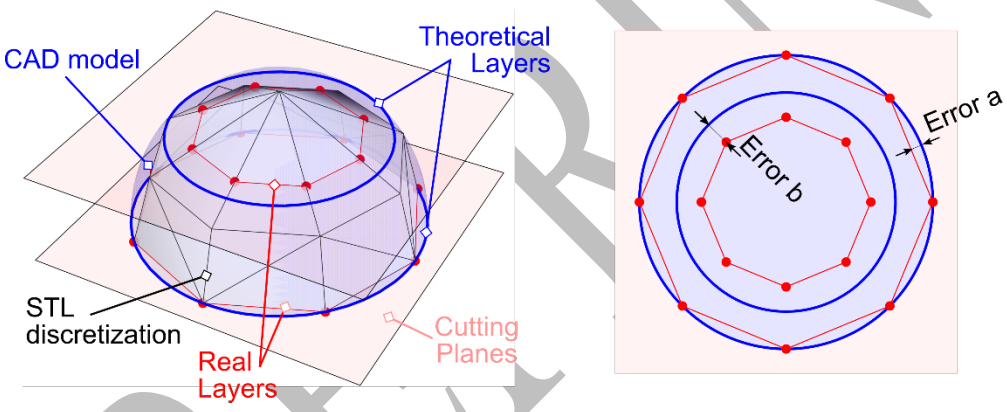

Fig. 1. Dimensional errors due to indirect slicing of a curvilinear surface.

On the other hand, the existing tracing and subdivision algorithms used to compute surface plane intersections allows direct slicing [22, 23]. Scholars claim that this approach reduces the mentioned errors [24]. Nevertheless, the intersection between a plane and a curvilinear surface for surface-based models is computationally more complex than the intersection between a plane and a triangular mesh [25]. In general, just like indirect slicing, direct slicing does not provide a curve but a set of intersection points, which should be used to generate a tool-path. In order to take advantage of the direct slicing to reduce error *a* (and the size of the instruction program), the idea is to approximate the intersections by polynomial curves.

Nevertheless, the resulting polynomial spline curves cannot be directly opened in the existing slicing software; thus, it is necessary to program specific applications to obtain the 3D printing instruction program. Additionally, to take advantage of the resulting analytical

curves' continuity and accuracy, a control is needed that allows interpolation schemes different from linear movements and writing the corresponding code in the instruction program. All these issues can hinder the application of direct lamination and explain that the indirect lamination of STL files is currently a standard step of AM processes [16].

This work proposes a blended approach that the authors call Direct Slicing Tessellation (DST). The idea is to slice the CAD model and use the intersection points to build a controlled STL representation. As a result, a reduction of type *b* error is expected compared to the conventional STL representation. Regarding error type *a*, it can be controlled by limiting the chord error. A program implemented with GrassHopper® is used to compute the intersections and build the DST. Due to DST leading to an STL file, the discrete model can be opened in conventional slicers and work as usual to guarantee integration in the development process.

Two parts, a hemispheric and a component of a retail product (a hair dryer nozzle), were modeled and processed in a conventional way and with our DST method. A reduction in dimensional error is predicted and noted for the hemispherical samples at different sections: the deviation between the path obtained from the G-code (for conventional and DST STLs) and the theoretical CAD model slices. On the other hand, for the printed components and using a 3D scanner to measure the resulting geometry, DST also shows lower dimensional and form deviations (considering the mean curvature and zebra stripes) than the conventional indirect slicing.

The paper is organized as follows: the first Section overviews the proposed slicing method. The second Section analyses the materials and methods used. The third Section examines the results obtained, and finally, the fourth Section exposes the conclusions.

2. Direct Slicing Tessellation (DST)

Instead of the conventional indirect slicing, where CAD model discretization is accomplished by controlling mainly the chordal tolerance and the angle control, the Direct Slicing Tessellation (DST) discretizes the CAD model considering type *a* and *b* errors.

DST approach consists of direct slicing at constant height steps of a NURBS CAD surface model and constructing a tessellation based on the obtained intersections. In this sense, error

b can be reduced to the numerical errors of the intersection algorithm, while error *a* can be controlled with the number of intersection points computed. Afterward, a mesh is produced from the intersection points and saved in STL format. Finally, the STL format is processed as usual in a slicing application.

Intersection stage.

In this study, Grasshopper®, a graphical algorithm editor within Rhinoceros® CAD software (McNeel & Associates Seattle, USA), is used to compute the plane-CAD model intersections. Regarding part surface–plane intersection, it is essential to consider how the printing software slices the surface. CURA is the slicing software used to produce the examples discussed in Section 4; this software has a middle slicing tolerance as a standard setting, i.e., slice at the middle of a layer (Fig. 2).

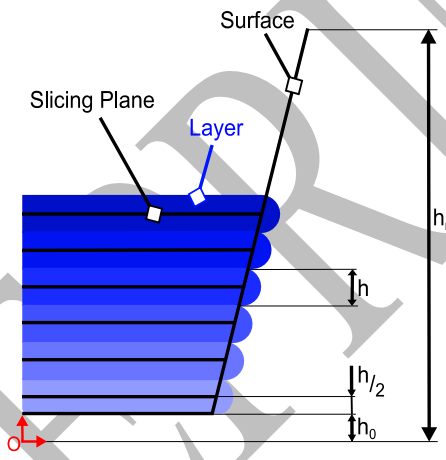

Fig. 2. Middle Slicing Tolerance according to Ultimaker Cura.

The layer height h defines the distance between consecutive intersection planes, except for the first slicing. Our algorithm places the first layer at $h/2$ from the printer bed to synchronize the intersections provided by the slicer software. Consequently, the following layers are placed at a distance h above the previous one. A simple python script was designed to achieve this goal (Fig. 3 and 4). The first slicing plane is generated at the minimum extent of the model h_0. Then, the second plane is generated at a distance $h_0 + h_i/2$. Finally, the rest of the planes are created by progressively adding a distance h_i until the maximum extension of the model h_n.

Tessellation stage.

The tessellation or discretization is also implemented in GrassHopper® (Fig. 3 and 4) and constructed from the resulting intersection points obtained from direct slicing. Specifying the maximum length between points is possible to define the number of intersection points that limit the error *a*. The optimal length will be the one that allows the best combination between the highest printing speed, the minimum error *a*, and the shortest computational time. For the experimental tests, the segment length is $l > 0.5$ mm to maintain the printing speed close to the programmed one [26], with a maximum value defined to obtain an error $a <$ 0.05 mm. Due to the previous definition, the maximum segment length in the examples is always < 1 mm; thus, it is used a minimum size of a segment after slicing of 0.83 mm in the CURA program. The resulting points are joined sequentially at each intersection, and a polyline is obtained. At last, a mesh is generated from these polygonal curves using the GrassHopper® plugin.

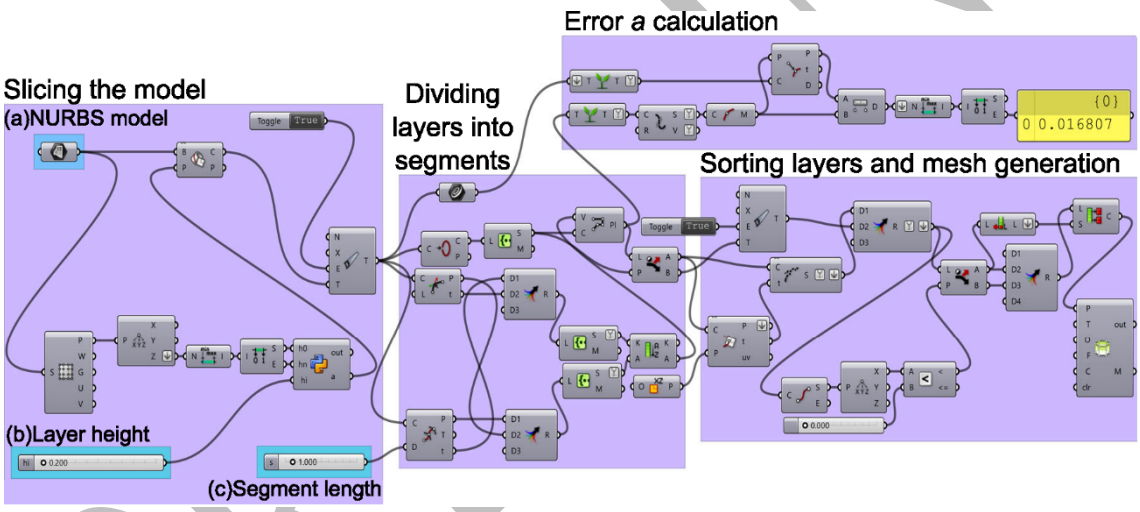

Fig. 3. Workflow of the DST method. The input data in this workflow are: (a) the NURBS model, (b) Layer height, and (c) Segment length.

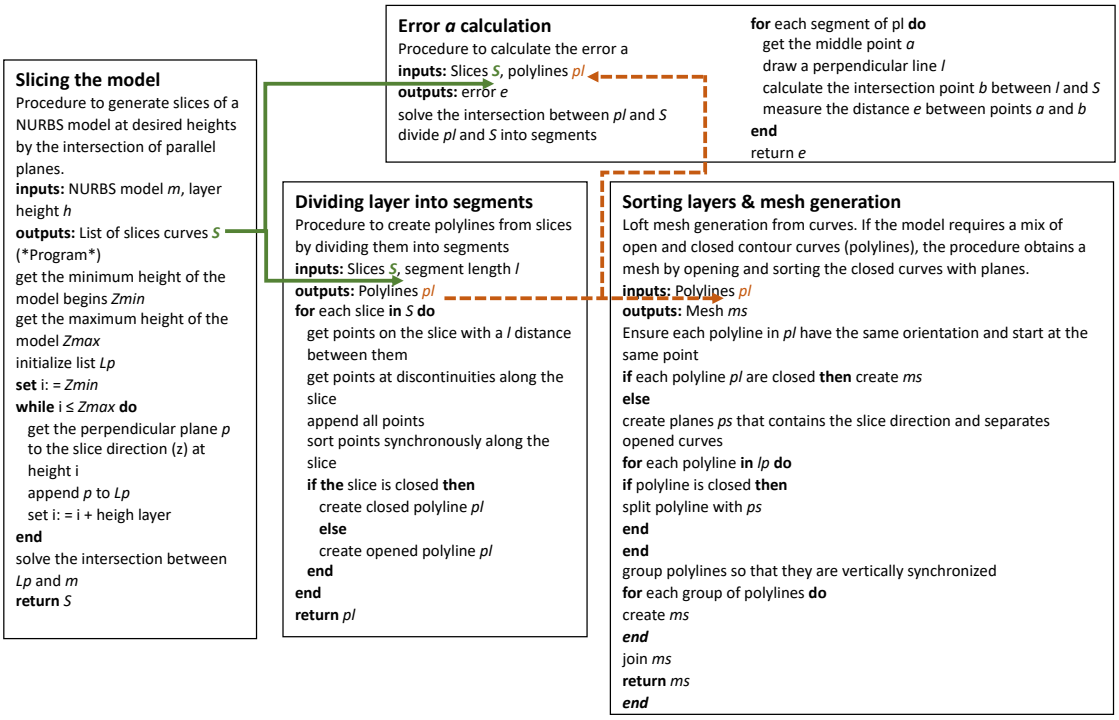

Fig. 4. Pseudocode of the program of blocks shown in Fig. 3.

3. Materials and Methods

Conventional STL tessellations and the proposed tessellation (Section 2) influence the resulting fidelity (dimensions and shape). This section describes the geometries printed and explains the tests accomplished to compare the conventional and proposed discretization.

Two curved 3D surfaces are defined and printed via Fused Filament Fabrication (FFF) to appreciate indirect slicing and DST differences. These examples are tested using theoretical estimation of deviations, a surface study based on zebra stripes, and 3D scanner measures.

3.1. Testing shapes and procedures

Hemispherical sample and numerical verification

For a hemispherical shape (Fig. 5), the intersections are known. Therefore, it is straightforward to quantify in advance the tessellation influence on the deviation between the CAD model and the geometry generated by the instruction program used for printing the part.

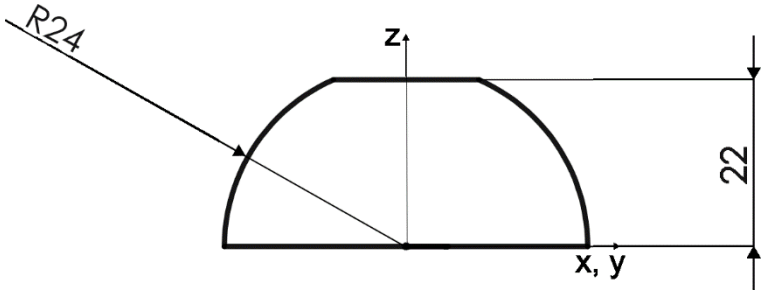

Fig. 5. Hemispherical geometry.

For a specific layer i, the nozzle travels through $\mathbf{p}_{i,j}$, j=1, 2,…, n_i points and the corresponding intersection with the hemispherical surface is a radius R_i circle. Therefore, the deviation δ_i is:

$$\delta_i = Max\left[\left|R_i - \left\|\mathbf{p}_{i,j}\right\|\right|\right]. \quad (1)$$

Moreover, for layer i, it is straightforward to compare the theoretical perimeter length and the instruction program (G-code) polygonal curve by computing the total distances between points l_j. The following equation is used to obtain the relative error e between the theoretical and programmed perimeter at layer i:

$$e_i = \left|2\pi R_i - \sum_{j=1}^{n_i-1} l_j\right| / 2\pi R_i. \quad (2)$$

The proposed DST is compared with three conventional discretization cases (Table 1). The STL meshes are obtained in Rhinoceros using the simple mesh control (maximum distance between model and mesh) considering, as reference for their definition, the configuration of the SolidWorks STL exporting profiles (fast, high quality, and highest quality).

The virtual-numerical tests performed with the hemispherical shape will aid in checking if the proposed tessellation can provide advantages over conventional discretization. Once this assumption is verified, the next step is to note the effect in a more complex or actual shape.

Real product component

Actual tests were based on producing a complex but common workpiece geometry: a dryer nozzle, which was modeled using NURBS (non-Uniform rational B-spline) [27] surfaces in Rhinoceros (Fig. 6). Its maximum dimensions are 87.89 × 47.17 × 61 mm. For this shape, the DST approach is compared with the conventional STL discretization, whose main characteristics are provided in Table 1. Note the significant reduction in mesh size and slicing time compared to the reference mesh with the highest resolution.

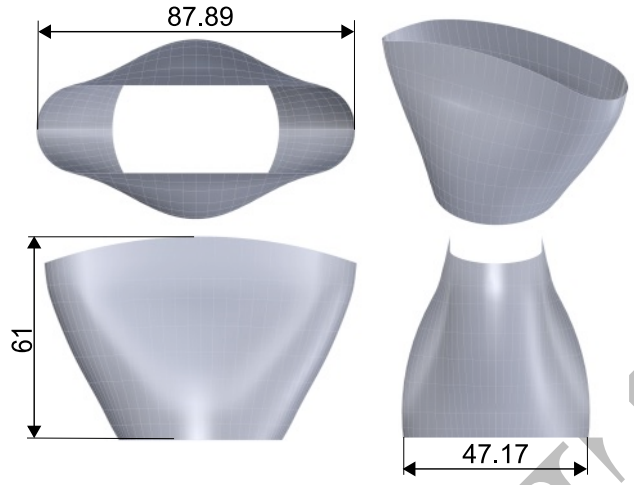

Fig. 6. Dryer nozzle geometry.

Table 1. Discretization cases of conventional STL and DST meshes compared in this study: size and saving time to the reference mesh.

	Tessellation	Tolerance / mm	Triangles	STL ASCII size	Saving time in slicing processing (%)
Hemisphere	Fast	0.085	886	246 kb	56
	High Quality	0.037	2814	785 kb	46
	DST	a error ≤ 0.025	19800	3.9 Mb	52
	Reference	0.00576	180178	50.3 Mb	0
Nozzle	Fast	0.1058	1838	612 kb	61
	DST	a error ≤ 0.0168	105720	29.5 Mb	40
	Reference	0.0003	783960	249 Mb	0

3.2. Printing process

Real samples were made using an Ultimaker S5 3D printer (Utrecht, Netherlands). Its main characteristics are a nozzle diameter of 0.4 mm, a build volume of 330 × 240 × 300 mm, and a minimum layer thickness of 20 µm. The printing material was PLA filament with a diameter of 2.85 mm.

Furthermore, the G-code (the program of instructions) was obtained from Ultimaker CURA software (a free and open-source LGPLv3 application developed for Ultimaker). Table 2 shows the process parameters defined in CURA for the hemispherical and nozzle samples. All samples were printed using PLA filament. We considered a layer height of 0.2 mm and

the default options, but for the maximum resolution (minimum size of a line segment after slicing), which was set to 0.83 mm, and the maximum deviation (deviation allowed for the maximum resolution) to 0.025 mm. These values provide a similar interpolation polyline at each slice than those obtained to build the DST tessellation. The authors verified, with several experimental prints, that to obtain the best surface quality, wall speed should be reduced accordingly to the reduction of the maximum resolution. In that sense, for the fast example, it is considered the wall speed corresponding to the CURA fast printing profile. On the other hand, the wall speed of *normal* and *fine* CURA profiles was used for the high-quality DST and reference samples, respectively. Finally, the Shell/*z* Seam Position was placed on the part's back. The default configuration of the CURA fast profile corresponding to a 0.2 mm layer height was taken for the rest of the parameters.

Table 2. Configuration of the printing process parameters using the slicing software CURA.

		Tessellation		
Category	**Process parameter**	**Fast**	**DST & High Quality**	**Reference**
	Max Resolution / mm		0.83	
Mesh fixes	Max Travel Resolution / mm		0.7	
	Max Deviation / mm		0.025	
Material	Printing Temperature / ºC	205	200	195
Speed Profiles	Wall Speed / (mm/s)	55	45	30

3.3. Measurement procedure

Model-shape dimensional deviation.

The printed hemispherical and nozzle shapes were compared to their CAD models using a 3D scanner. The scanner employed was the EinScan SP model manufactured by Shining 3D, which uses its own software (EXScan S v.3.1.0.1.). The scanner is based on structured light (SL), i.e., it uses a light source that projects different patterns on the object, which are captured by a digital camera to recover the geometry of the solid based on the deformations of the light patterns. Then, with the help of a mold made of silicon using the reference sample, the object was placed in a built-in rotating base to define its geometry fully (360º scan). The scanner has a point cloud resolution of between 0.17 and 0.2 mm and a scanning uncertainty of ≤ 0.05 mm.

The point cloud obtained from each scanned object was processed and compared to the CAD model's finest discretization (reference) using the CloudCompare open-source software [28]. The analysis of the 3D surface deviation between the reference and scanned model requires that both clouds are aligned. This is why the well-known Iterative Closest Point algorithm (ICP) was used for the registration by CloudCompare. The process is based on the original ICP method, introduced by Besl and McKay [29], with some improvements [30], highlighting the convergence speeding by randomly subsampling the aligned cloud at each iteration [31]. The ICP registration process stops when the RMS difference attains 10^{-6} to achieve a fine combination between convergence time and accuracy. Besides, an overlap between each point of the reference cloud and the scanned model was set to 100% to achieve the highest possible accuracy and a random size of 150,000 points (close to the volume of the scanned models) to refine the registration. We do not adjust the scale considering that scans capture the actual size of the samples and that we want to measure the variation of the printing.

After the alignment step, the two clouds were compared using the cloud-to-mesh (C2M) tool of CloudCompare. The algorithm returns the distances between every point of the compared cloud (scanned model) and the nearest triangle of the reference mesh. When the orthogonal projection of a given point lies within the triangle, the distance is between that point and its intersection on the triangle. Otherwise, the distance will be between the point and the nearest edge of the triangle [32]. Moreover, it is known what is inside and outside the mesh by looking at the normal of each triangle. It should be noted that we only compared half of each sample without considering the seam in order to study the smooth surface without imperfections.

Surface quality analysis.

Additionally, for the Nozzle samples, it is also evaluated the 3D surface quality by computing the mesh curvature and the zebra stripes. The mean curvature (**H**) is evaluated using CloudCompare, while the surface continuity is examined with the zebra striping diagnostic shading technique [33] using Rhinoceros 3D CAD software.

Regarding **H** computation, for a given point and its neighbors, the CloudCompare application computes the quadric surface which best fits them and deduces the curvature value for the central point, as suggested by Har'el [34]:

$$H = \frac{(1 + f_x^2)f_{yy} - 2f_x f_y f_{xy} + (1 + f_y^2)f_{xx}}{2(1 + f_x^2 + f_y^2)^{\frac{3}{2}}} \quad (3)$$

Determining the range of neighboring points is critical in calculating the abovementioned parameters. On the one hand, the greater the number of neighboring points, the less sensitive the measurement will be to noise. On the other hand, if there are few neighboring points, it is required to ensure that the data is noiseless and that there are enough points to estimate the parameter. In CloudCompare software, it is possible to control the number of neighbors by setting the radius of a sphere inside which the neighbors are extracted around each point. In this research, the radius that best describes the clouds without losing key points was 1.8 mm. It is known that nine points in three-dimensional Euclidean space determine a unique quadric that passes through those points. So, the minimum possible value for the radius is the one whose sphere contains nine points inside. The maximum value for the radius was detected visually by comparing the scanned surface with the real one and observing when the imperfections of the surface were no longer visible.

4. Results and Discussion.

Hemispherical samples

In order to study the impact of the proposed DST method, we accomplished a comparison based on the slicing codes and a comparison of the printed hemispherical samples measured using a 3D scanner.

Regarding the virtual evaluation, we compare along the building *z* direction the G-code obtained for the cases shown in Table 1 using the configurations described in Table 2 and the theoretical intersection curves of the hemisphere. It is possible to acquire the polygonal paths along *z* height from the instruction program generated from the sliced tessellation. These paths are similar to the examples shown in Fig. 1.

According to the results in Fig. 7, it stands to reason that in conventional tessellations, the highest deviations occur for *z* slices where tessellations vertices are the farthest from the intersection. The lowest deviations happen when those vertices are close to the intersection. The larger the number of triangles, the more times the minimum values occurred, as shown in Fig. 7. This result agrees with the error *b* defined in Section 1.

In the limit for the highest number of triangles, the deviation is uniform and has a constant value of ~0.175 mm. The actual deposition is not a line; there is a strip with a thickness equal to the nozzle diameter and a height equal to the layer height. Hence, the actual path defined in the instruction program should be the offset of the theoretical intersection curve at a distance equal to half the nozzle diameter. For a nozzle with a 0.4 mm diameter, the line width programmed by default in CURA is 0.35 mm, corresponding with an offset distance of 0.175 mm, the same value observed in Fig. 7 (Left).

Therefore, considering that the adequate positions are those defined by the offset mentioned above, the proposed DST method and the reference provide the most accurate dimensional results. However, the reference tessellation file size is more than ten times greater than the proposed DST method.

Regarding the relative boundary length error (Fig. 7, Right), the DST approach is again similar to the reference, and both bound the minimum values of fast and high-quality tessellations. Besides, the profiles show low values, and the oscillations observed attend to the same reasons discussed previously. On the other hand, as z grows, the number of intersection points in each layer decreases, increasing the error a (Fig. 1) and explaining the no decreasing monotonic behavior observed.

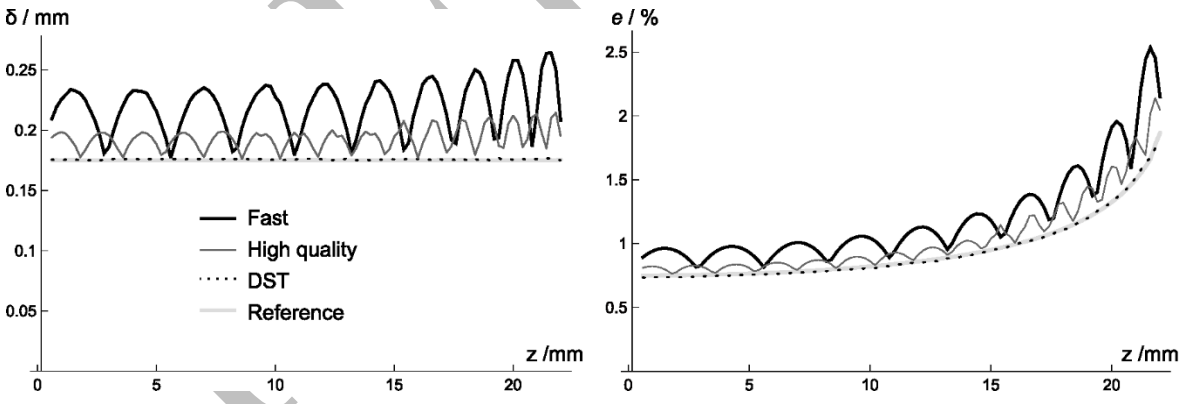

Fig. 7. Theoretical maximum deviation (left) and boundary length error (right) through the building direction z.

According to the virtual results, a better appearance for the DST approach is expected compared with the fast and high-quality samples. The photographs of the real examples corroborate that hypothesis (Fig. 8).

Additionally, Fig. 9 shows the deviations to the reference of the samples' dimensions measured by 3D scanning. This figure reveals that the DST surface shape agrees with the reference and shows the lowest maximum deviation.

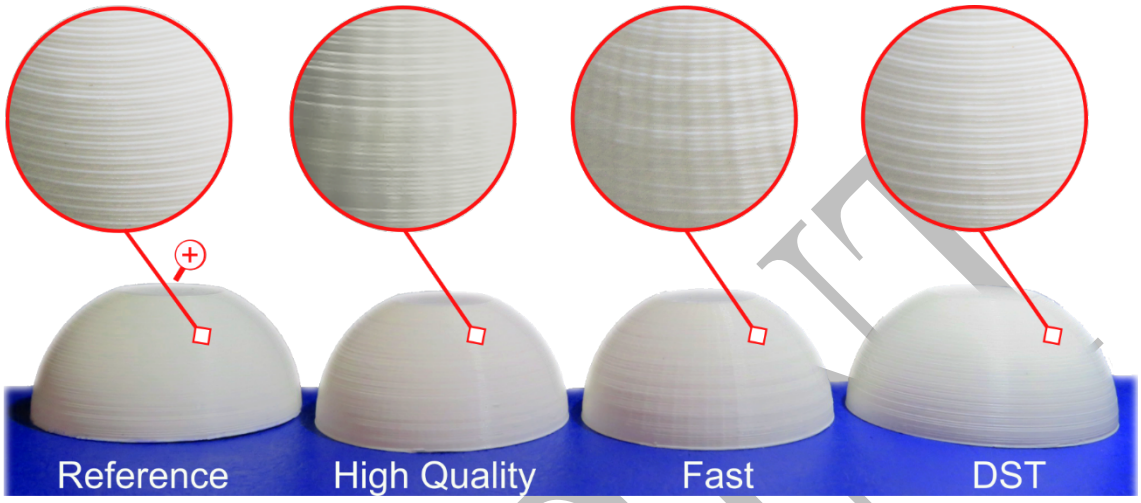

Fig. 8. Printed hemispherical samples with surface appearance details.

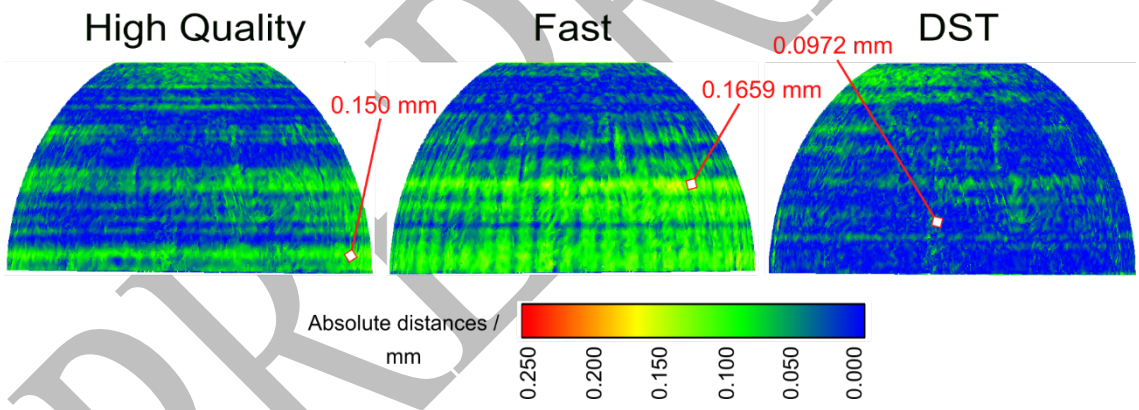

Fig. 9. Dimensional deviations to the reference.

Nozzle samples

Regarding the appearance of the nozzle samples (Fig. 10), it can see that the reference and DST surfaces are similar and appear to be smoother than the fast surface. In all cases, the layers are visible due to the stair-stepping effect. However, on the fast surface, the mesh is also visible.

As in the previous discussion, the deviation analysis (Fig. 11) shows that the fast surface shape fits worse to the reference; it has greater and more dispersed dimensional deviations, easily seen in green and red. This representation allows visualizing the discretization in the

printed fast sample. On the other hand, the DST surface has better accuracy and only shows significant deviation around the center. Moreover, it has a lower maximum deviation than the fast surface.

Concerning mean curvature (Fig. 12), it is easy to see that there are no main differences between the reference and DST surfaces. At the same time, the fast one matches the reference worse due to the mesh being noticeable.

Finally, the results of zebra patterns (Fig. 13) reveal that the reference and DST surfaces are similar. In addition, their surfaces are smoother than the fast surface, which has several visual discontinuities (observe its broken stripes).

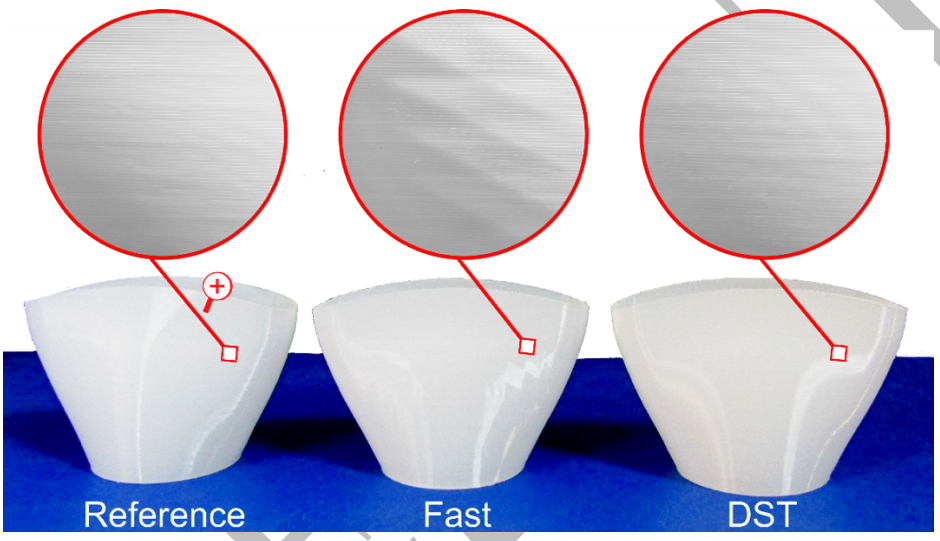

Fig. 10. Printed nozzle samples with surface appearance details.

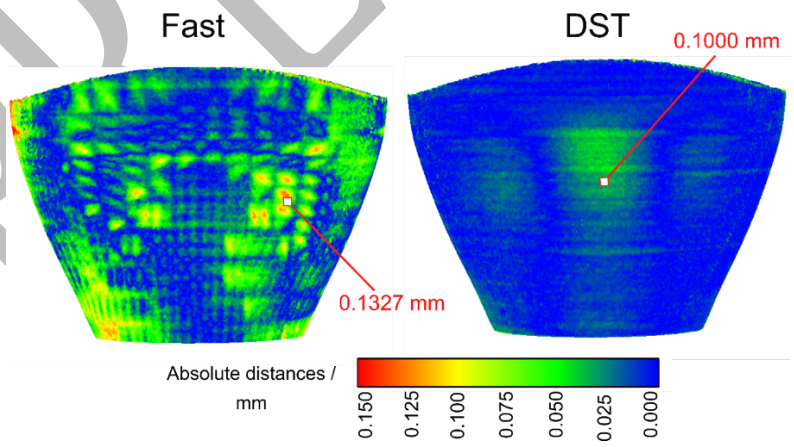

Fig. 11. Dimensional deviations to the reference.

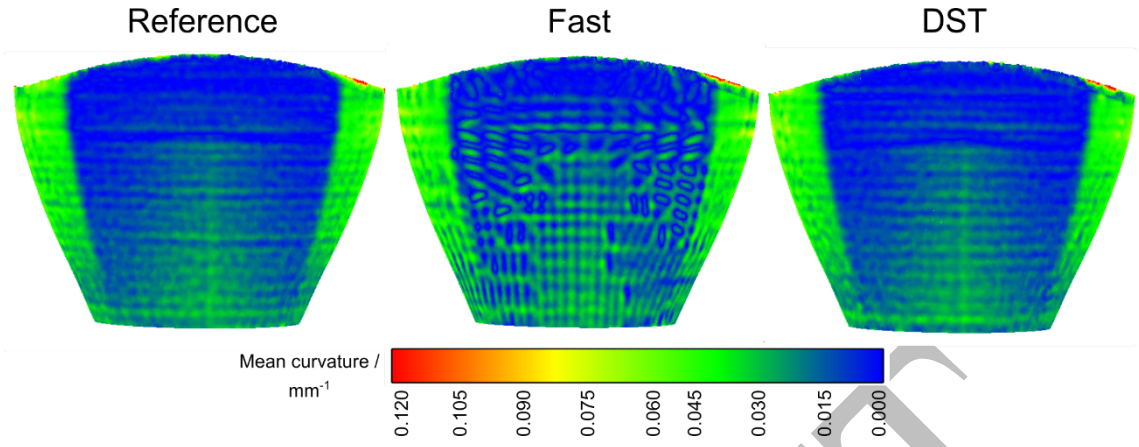

Fig. 12. Mean curvature color map.

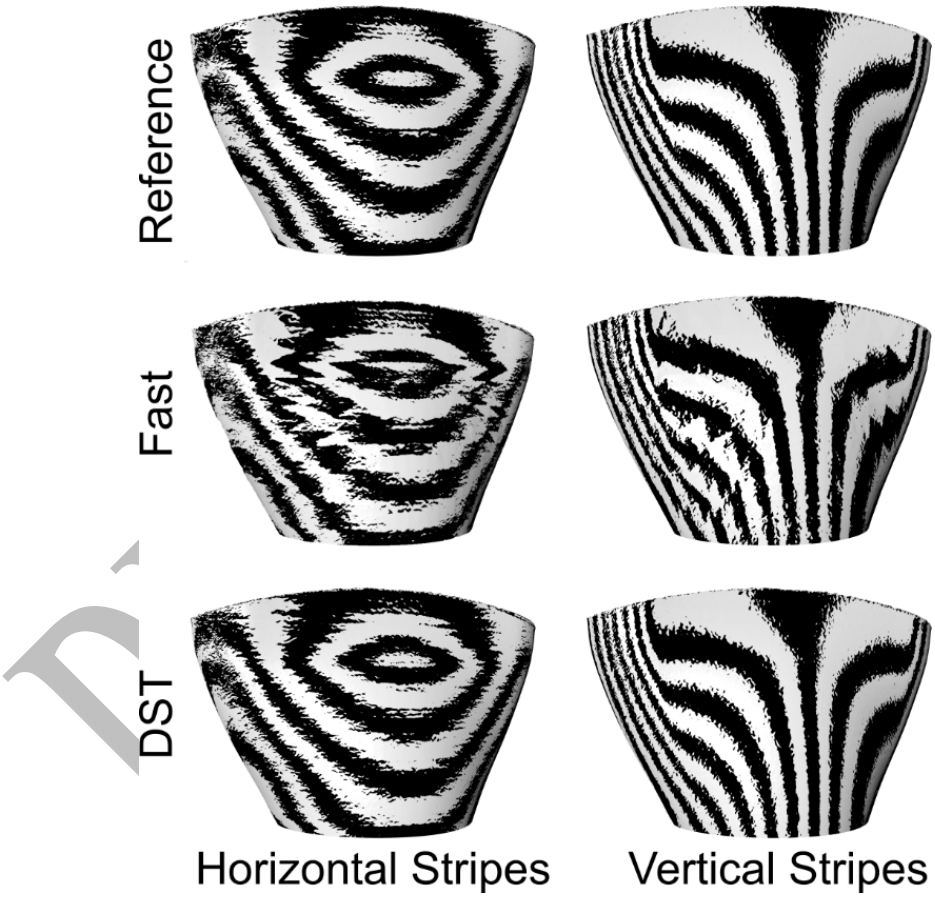

Fig. 13. Zebra stripes pattern on printed nozzle measured meshes.

5. Conclusions

In additive manufacturing processes, the geometrical operations required to cut curvilinear CAD models in 2D layers are a source of errors for the final printed surface. The

conventional procedure is based on saving a discretization approximation of the CAD model in STL format and a slicing procedure using parallel planes. However, this solution shows in each 2D layer an error (type *a*) between the theoretical intersection curve and the polygonal one. Moreover, there is an error (type *b*) in the printing direction when the cutting planes intersect triangle faces instead of STL vertices.

Building a fine mesh to reduce the errors mentioned above is possible. However, that increases the STL file size and the slicing's computational time and can lead to significant printing times when the tool path is defined by G1 instructions (linear movements). Another option is cutting the CAD model instead of the STL file. Direct slicing eliminates type *b* error; the intersection algorithms already implemented inside CAD software can aid in performing this type of slicing. However, how to transform the intersections into the instruction program required to print a part with reduced type a and b errors is complex and needs a specific hardware configuration.

The procedure proposed in this work, which we call Direct Slicing Tessellation (DST), builds an STL file using the intersections with the CAD model. The paper explains how to implement the intersections and the STL construction using Grasshopper®. The developed function allows selecting the cutting direction and the distance between cutting planes and controlling the maximum chord length between intersection points.

The slicer programs can open and process the improved STL files as usual. It is just needed to maintain the same slicing direction and a layer height equal to the cutting planes' distances. Then, the slicer program can generate the instruction program without additional programming efforts.

Two sets of examples were printed: a hemisphere and a hairdresser nozzle, both with double-curved surfaces. The numerical study conducted for the hemisphere exposed the agreement of the DST with the theoretical surface, and the effect of error *b* in the deviation oscillations showed by the conventional STL tessellations. Finally, the measurements performed on the real printed hemispheres substantiate the numerical observations.

Regarding the printed nozzle, the observed deviations are in the same line as the hemispherical examples: DST shows a uniform deviation distribution and lower maximum deviation compared to conventional STL discretization, serving the results as a validation of the presented technique. Furthermore, the study of the nozzle is completed using tools to test the fidelity of the measured printed surfaces: mean curvature and zebra stripes. The study

shows that these tools also help to compare the resulting appearance with straightforward 3D scanning measurements.

The computed zebra stripes and mean curvature on the scanned point cloud show a better agreement with the CAD model of the DST parts than the conventional STL ones. Additionally, it is noted that the DST tessellation file size is ten times lower and around half the slicing time than the reference STL file obtained with the best resolution.

Although the examples were produced using FFF, the approach can be extended to enhance any AM process based on indirect tessellation and be integrated into its pipeline. Furthermore, the validation by numerical and scanned measurements described in this paper can help further investigations regarding the effects of slicing on dimensional and shape deviations.

References

1. ISO/ASTM 52900. ISO/ASTM 52900:2015. Additive manufacturing - General principles - Terminology. (2015).

2. Hague, R., Dickens, P. & Hopkinson, N. in *Rapid manufacturing: an industrial revolution for the digital age* (John Wiley & Sons, 2006).

3. Seleznev, M. & Roy-Mayhew, J. D. Bi-metal composite material for plastic injection molding tooling applications via fused filament fabrication process. *Additive Manufacturing* **48**, 102375 (2021).

4. Kading, B. & Straub, J. Utilizing in-situ resources and 3D printing structures for a manned Mars mission. *Acta Astronaut.* **107**, 317-326 (2015).

5. Baturynska, I. Statistical analysis of dimensional accuracy in additive manufacturing considering STL model properties. *The International Journal of Advanced Manufacturing Technology* **97**, 2835-2849 (2018).

6. Pérez, M., Medina-Sánchez, G., García-Collado, A., Gupta, M. & Carou, D. Surface quality enhancement of fused deposition modeling (FDM) printed samples based on the selection of critical printing parameters. *Materials* **11**, 1382 (2018).

7. Koch, C., Van Hulle, L. & Rudolph, N. Investigation of mechanical anisotropy of the fused filament fabrication process via customized tool path generation. *Additive Manufacturing* **16**, 138-145 (2017).

8. García-Collado, A., Romero-Carrillo, P. E., Dorado-Vicente, R. & Gupta, M. K. Studying the Effect of Short Carbon Fiber on Fused Filament Fabrication Parts Roughness via Machine Learning. *3D Printing and Additive Manufacturing* (2022).

9. Vyavahare, S., Kumar, S. & Panghal, D. Experimental study of surface roughness, dimensional accuracy and time of fabrication of parts produced by fused deposition modelling. *Rapid Prototyping Journal* **26**, 1535-1554 (2020).

10. McGregor, D. J. *et al*. Analyzing part accuracy and sources of variability for additively manufactured lattice parts made on multiple printers. *Additive Manufacturing* **40**, 101924 (2021).

11. Baturynska, I. & Martinsen, K. Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *J. Intell. Manuf.* **32**, 179-200 (2021).

12. McGregor, D. J., Bimrose, M. V., Shao, C., Tawfick, S. & King, W. P. Using machine learning to predict dimensions and qualify diverse part designs across multiple additive machines and materials. *Additive Manufacturing* **55**, 102848 (2022).

13. Pandey, P. M., Reddy, N. V. & Dhande, S. G. Slicing procedures in layered manufacturing: a review. *Rapid prototyping journal* **9**, 274-288 (2003).

14. Zhang, Y. *et al*. A systematic review on data of additive manufacturing for machine learning applications: the data quality, type, preprocessing, and management. *J. Intell. Manuf.*, 1-36 (2022).

15. Zhao, Z. & Luc, Z. Adaptive direct slicing of the solid model for rapid prototyping. *Int J Prod Res* **38**, 69-83 (2000).

16. Attene, M. *et al*. Design, representations, and processing for additive manufacturing. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging* **10**, 1-146 (2018).

17. Leary, M. in *Chapter 3: Design for additive manufacturing.* (Elsevier, Amsterdam, 2019).

18. Munteanu, A. *et al*. A Study on the Errors of 2D Circular Trajectories Generated on a 3D Printer. *Applied Sciences* **11**, 11695 (2021).

19. Hällgren, S., Pejryd, L. & Ekengren, J. 3D data export for additive manufacturing-improving geometric accuracy. *Procedia Cirp* **50**, 518-523 (2016).

20. Qin, Y., Qi, Q., Scott, P. J. & Jiang, X. Status, comparison, and future of the representations of additive manufacturing data. *Comput. -Aided Des.* **111**, 44-64 (2019).

21. Oropallo, W. & Piegl, L. A. Ten challenges in 3D printing. *Engineering with Computers* **32**, 135-148 (2016).

22. Lee, K. H. & Woo, H. Direct integration of reverse engineering and rapid prototyping. *Comput. Ind. Eng.* **38**, 21-38 (2000).

23. Alkadi, F., Lee, K., Bashiri, A. H. & Choi, J. Conformal additive manufacturing using a direct-print process. *Additive Manufacturing* **32**, 100975 (2020).

24. Nayyeri, P., Zareinia, K. & Bougherara, H. Planar and nonplanar slicing algorithms for fused deposition modeling technology: A critical review. *The International Journal of Advanced Manufacturing Technology* **119**, 2785-2810 (2022).

25. Starly, B., Lau, A., Sun, W., Lau, W. & Bradbury, T. Direct slicing of STEP based NURBS models for layered manufacturing. *Comput. -Aided Des.* **37**, 387-397 (2005).

26. Medina-Sanchez, G., Dorado-Vicente, R., Torres-Jiménez, E. & López-García, R. Build time estimation for fused filament fabrication via average printing speed. *Materials* **12**, 3982 (2019).

27. Piegl, L. & Tiller, W. in *The NURBS book* (Springer Science & Business Media, Berlin, 1996).

28. http://www.cloudcompare.org/ [accessed 14 April 2021].

29. Besl, P. J. & McKay, N. D. Method for registration of 3-D shapes. *Proc. SPIE 1611, Sensor Fusion IV: Control Paradigms and Data Structures* **1611**, 586-606 (1992).

30. Rusinkiewicz, S. &Levoy, M. *Efficient variants of the ICP algorithm* (Proceedings third international conference on 3-D digital imaging and modeling, IEEE, 2001).

31. Masuda, T., Sakaue, K. & Yokoya, N. *Registration and integration of multiple range images for 3-D model construction* (Proceedings of 13th international conference on pattern recognition Ser. 1, IEEE, 1996).

32. Girardeau-Montaut, D. Détection de changement sur des données géométriques tridimensionnelles. *Télécom ParisTech* (2006).

33. Pellis, D. et al. *Architectural freeform surfaces designed for cost- effective paneling through mold re-use* (*Proceedings of the Advances in Architectural Geometry*, 2020).

34. Har'el, Z. Curvature of curves and surfaces–a parabolic approach. *Department of Mathematics, Technion–Israel Institute of Technology* (1995).

Acknowledgements

Grant PID2019-104586RB-I00 funded by MCIN/AEI/10.13039/501100011033 supported this work. In addition, the authors thank the Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía for the financial support obtained through the Project (ProyExcel_00662) within the framework of the FEDER-Andalucía 2014-2020 program.