Additive Manufacturing 35 (2020) 101254

Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma

Support-free hollowing with spheroids and efficient 3D printing utilizing circular printing motions based on Voronoi diagrams

Seunghwan Choi[a], Joonghyun Ryu[b], Mokwon Lee[a], Jehyun Cha[a], Hyunwoo Kim[a], Chanyoung Song[a], Deok-Soo Kim[a,b,*]

[a] School of Mechanical Engineering, Hanyang University, Republic of Korea
[b] Voronoi Diagram Research Center/HYU-HPSTAR-CIS High Pressure Research Center, Hanyang University, Republic of Korea

ARTICLE INFO

Keywords:
3D printing
Overhang
Packing
Tool path
Circular motion

ABSTRACT

Extrusion-based printing frequently requires a hollowing step to remove material from inside of artifacts and subsequently reduce the amount of material, printing time, product weight, energy consumption, and ultimately, the cost. Here, we introduce a novel support-free hollowing method that uses an arrangement of vertically-aligned prolate spheroids. In addition to reducing stress concentration through their inherently smooth boundaries, these spheroids require no additional support structure, when properly designed. Additionally, the resulting spheroidal hollows facilitate the circular printing motion of extruders using G2/G3-codes, which provide three critical advantages compared to the currently popular G1-code-based linear motion: shorter printing time, better printing quality, and smaller tool path file. Here, spheroids are arranged by the Voronoi diagram of 3D ellipsoids and the tool path, including circular printing motions, is produced using the Voronoi diagram of circular 2D disks. The proposed algorithms are implemented as the HollowTron webserver and are freely available from Voronoi Diagram Research Center.

1. Introduction

Additive manufacturing (AM), also frequently referred to as 3D printing, has become popular due to its high degree of freedom for producing the geometric and functional complexities of various products that could not otherwise be achieved. To implement AM, extrusion-based printing technologies, such as fused deposition modeling and fused filament fabrication, are the most popular among the diverse array of existing AM technologies.

Unfortunately, extrusion-based printing frequently requires additional support structures to prevent relatively large horizontal overhang parts from falling during printing [1–3]. Typically, there must be material below or near to the printing location in the current layer, or gravity can call the newly printed material to fall. These extra support structures have to be removed, either manually or by dissolving material away from the printed objects after printing is finished. Notably, support structures may be used either in the exterior or interior of a product model, and exterior structures, like scaffolds, can be removed much more easily, with a low cost of time and effort. Thus, in addition to generating excess material waste, these support structures, particularly internal ones, increase the production and post-processing times, so support structures have an immediate influence on the productivity and planning for AM processes.

Hollowing is a general technique used to eliminate material from the interior of printed objects. Extrusion-based printing frequently requires a hollowing step to reduce the amount of material used, printing time, product weight, energy consumption, and overall cost [4]. Additionally, hollowing can be used to control the material distribution and subsequently alleviate stress at certain object locations. In applications, hollows can be used to host living organisms [5], to design electrode materials with high specific capacity [6] and dielectric structures [7], and serve various other purposes.

However, if a hollow is not carefully designed, it may have a long overhang, beyond the technical limit of the material printing and cooling system. Such structures require additional interior support structures, which can only be removed after printing, by breaking the boundary of printed object. Some studies have successfully created support-free interior vacancies by satisfying both overhang length and boundary slope constraints [8–12] by filling an interior hollow with a self-supporting infill pattern that can effectively support hollow ceilings. Some landmark studies in this vein include a honeycomb-cell hollowing with a relatively smooth boundary [13], an elliptic cylinder

* Corresponding author at: School of Mechanical Engineering, Hanyang University, Republic of Korea.
E-mail address: dskim@hanyang.ac.kr (D.-S. Kim).

https://doi.org/10.1016/j.addma.2020.101254
Received 9 December 2019; Received in revised form 27 March 2020; Accepted 13 April 2020
Available online 12 May 2020

representation of hollows [14], using a rhombic cell structure as an infill pattern, where the slope angle of all rhombic cells was constrained by a prescribed maximum overhang-angle [15], and employing block representation of hollows using partitioned blocks [16]. However, each of these methods has C^0-continuity, i.e. a sharp corner, at some locations on the hollow boundary, which inherently create a high stress concentration [17] and a higher crack development probability. Further details are provided in [14]. A thorough literature review on hollowing and support-free printing structures is provided in Appendix A. The relatively simple idea of rounding sharp corners can significantly increase the geometric complexity of an object, thus increasing the size of CAD data files and the numerical instability of algorithms. Notably, the proposed hollowing method can also be useful for the digital light processing based 3D printing [18–21].

In the typical process, an object is represented in a CAD model and then printed by a 3D printer. The printer is a numerically controlled machine and its extruder, or printer head, is driven by a tool path that contains G-codes that correspond to extruder motions. The four G-code types for different motions are G0, G1, G2, and G3, which correspond to rapid positioning, linear motion, clockwise circular motion, and counterclockwise circular motion, respectively. Currently, most printers use only G1-codes, in addition to G0-codes used for positioning. There are two main reasons for this. First, a CAD model is stored in an STL file as a set of triangles. This makes it a challenge to correctly extract abstract information such as circularity, sphericity, etc. Second, an interpolator for G2/G3-codes is more expensive than that for G1-code [22,23]. Importantly, circular motions offer three significant advantages: shorter printing times, better printing quality, and smaller tool path files. Appendix B provides brief review of tool path generation for printing.

Ourwork. Here, we introduce novel methods for (i) support-free hollowing and (ii) printing artifacts using as many circular motions, with G2/G3-codes, as possible. First, the support-free hollowing is produced by arranging vertically aligned 3D prolate spheroids in the object CAD model. Importantly, the properly designed spheroidal hollow can be support-free and experience a significantly reduced stress concentration. In this study, spheroids are defined as mutually exclusive and placed to maintain a prescribed separation distance threshold, both between adjacent spheroids and between spheroids and the object boundary. Additionally, the size and shape of each spheroid depends on the material stickiness property combined with the cooling parameters of the specific printing facility. Second, spheroidal hollows are printed using circular motions with G2/G3-codes, facilitating an improved printing quality and efficiency. Each printing layer is defined by the intersection between the object and a plane orthogonal to the printing direction. Typically, this plane is parallel to the XY-plane, because the printing direction corresponds to the Z-axis. Note that, in this study, the object contains vertically-aligned spheroidal hollows. Therefore, the printing layer defines a polygon to fill material, but circular regions are maintained as empty, thus this feature facilitates fewer circular G2/G3-codes rather than many short linear G1-codes.

The 3D spheroids are arranged using the Voronoi diagram of spheroids which is based on the Voronoi diagram of 3D spherical balls. Starting with the Voronoi diagram of the interior of a polyhedron boundary, a new incrementing spheroid, at the Voronoi vertex with the maximum clearance, can be inserted individually in a greedy manner. Once a new spheroid is placed, the Voronoi diagram is updated with the spheroid, and the spheroid increment process is repeated as many times as necessary. The key challenge here is in correctly maintaining the Voronoi diagram of spheroids within a polyhedral CAD model. This study is an extension of our recent work [14], which introduced the concept of support-free hollowing using smooth surfaces. The tool path with G2/G3-codes is generated using the Voronoi diagram of circular disks in a 2D polygon, where a disk is derived from a spheroid and the polygon from the boundary of CAD model. A thorough numerical simulation was conducted to verify and validate the proposed method, by performing stress and deformation analyses using the ANSYS® program. Artifacts were then actually printed using a SINDOH 3DWOX 1X® printer. **Contributions**. The notable contributions of this work are summarized in the following developments:

1. A method for generating support-free spheroidal hollows in a polyhedral object.
2. A method for generating tool paths using circular printing motions.
3. The HollowTron webserver, which is freely available at http://voronoi.hanyang.ac.kr/hollowtron.

Notes. In this study, an object to print is assumed to be represented as an oriented 2-manifold triangular mesh. The build, or printing, orientation is parallel to the Z-axis. In fact, it can be specified either manually or by an automatic algorithm such as [24]. “Object”, “model”, and “artifact” are interchangeably used, depending on context. An object is a connected solid and is printed all at once. Specifically, no consideration is given for: (i) the decomposition of an object into a set of primitive objects, which can be assembled later in a support-free manner [25], or (ii) multi-directional printing using a rotational platform [26]. “Hollow” refers to a designed vacant space, whereas “void” refers to an unintentional non-designed vacant space in a printed artifact. “V-vertex” denotes a vertex of Voronoi diagram, and “V-edge,” “V-face,” and “V-cell” are used similarly.

2. Material and methods

2.1. Modeling support-free spheroids

Fig. 1 shows a 2D schematic model of the printing process for a vertically-aligned elliptic hollow on the XZ-plane. The printing direction is assumed to be parallel to the Z-axis. The green slabs denote printing layers.

2.1.1. Printing parameters

Let σ_0 be the thickness of each printing layer which is 0.1–0.4 mm for general extrusion-based printers (Fig. 1(a)). Notably, let δ_0 be the **maximum overhang length** that can be successfully printed without an additional support structure. Further, let θ_0 be the **maximum overhang angle** (Fig. 1(b)). Notably, σ_0 is printer-dependent, while δ_0 and θ_0 depend on both the printing material properties, such as stickiness, and on the printing environment, including the temperature and presence of wind. Typically, the parameters for plastic PLA (polylactic acid) materials are as follows: $\delta_0 = 5$ mm and $\theta_0 = 60°$.

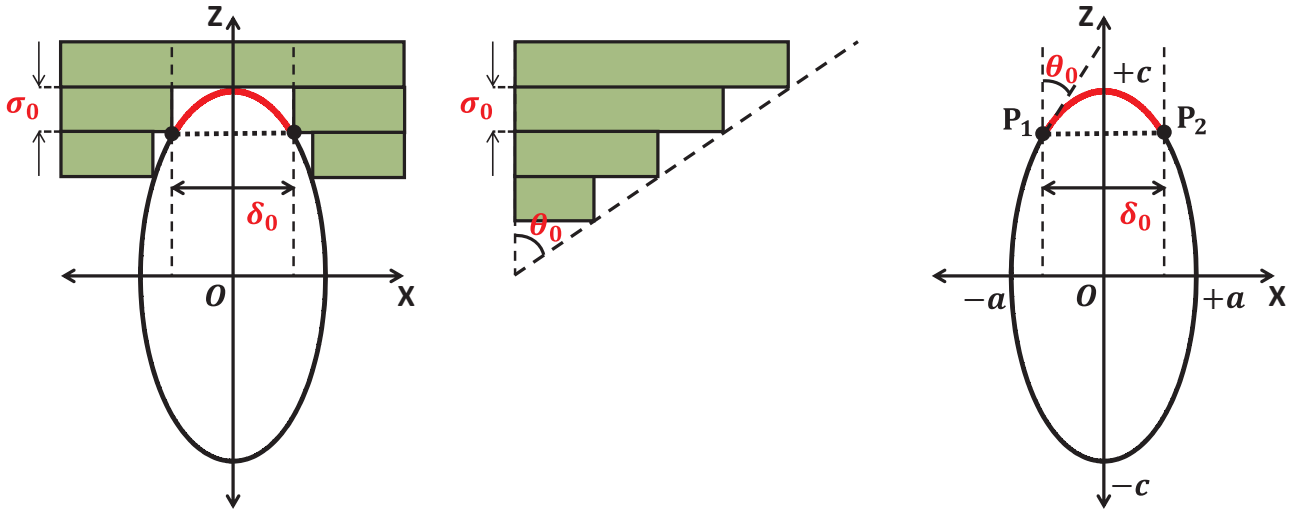
Let τ_{btw} and τ_{bndry} be the minimum wall thickness between adjacent spheroids and between a spheroid and the model boundary, respectively. For example, $\tau_{\text{btw}} = \delta_0$ and $\tau_{\text{bndry}} = 2\delta_0$. Having two different thresholds is desirable because an object boundary can be exposed to sharp impacts from the environment. In general, $\tau_{\text{bndry}} \geq \tau_{\text{btw}} \geq \delta_0 > \sigma_0$. The lower bound on the spheroidal hollow size is determined by the machine resolution, printing material properties, and design specifications, among other specifications.

2.1.2. Equations of ellipsoid and spheroid

An ellipsoid equation is given as:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{1}$$

where a, b, and c are the half lengths of the ellipsoid principal axes on the X-, Y-, and Z-axes, respectively. In this research, we used prolate spheroids, which are ellipsoids with $a = b < c$ and with the major axes parallel to the Z-axis, because the printing direction is assumed to be parallel to the Z-axis. A prolate spheroid is obtained by revolving an ellipse around its major axis. Ignoring the second term in Eq. (1) yields an ellipse equation, shown on the XZ-plane (Fig. 1(c)) as:



(a) (b) (c)

Fig. 1. A 2D model of a spheroidal hollow (shown as an ellipse in 2D) as it relates to the printing parameters of layer thickness, horizontal overhang length, and overhang angle. The printing direction is parallel to the Z-axis, and green slabs represent printing layers. (a) maximum overhang length δ_0. (b) Maximum allowed overhang angle θ_0. (c) Relationship between δ_0 and θ_0 for producing the widest possible ellipse.

$$\frac{x^2}{a^2} + \frac{z^2}{c^2} = 1. \tag{2}$$

Let P_1 and P_2 in Fig. 1(c) be the points on the ellipse where the tangent lines correspond to θ_0. The following simple, yet important, observation can then be made: If the distance between P_1 and P_2 is equal to or shorter than δ_0, the entire red elliptic arc can be printed by one printer motion stroke, which produces a straight linear path segment in the layer being printed. Thus, the red elliptic arc between P_1 and P_2 can be printed without any extra support, even though the overhang angle of the points inside the arc are greater than θ_0. Consequently, the entire ellipse can be printed without any additional supports.

The slope of the tangent line of Eq. (2) ellipse, at a point $p(x, z)$, is $-\frac{c^2}{a^2} \times \frac{x}{z}$. Then, P_1 and P_2 correspond to the solutions of the following equation:

$$-\frac{c^2}{a^2}\frac{x}{z} = \tan\left(\frac{\pi}{2} - \theta_0\right) \tag{3}$$

Substituting Eq. (2) into Eq. (3) yields two solutions for x: $-\frac{a^2}{\sqrt{c^2\tan\theta_0{}^2 + a^2}}$ and $\frac{a^2}{\sqrt{c^2\tan\theta_0{}^2 + a^2}}$. Given $\frac{2a^2}{\sqrt{c^2\tan\theta_0{}^2 + a^2}} \le \delta_0$, the bound of a is as follows:

$$\frac{\delta_0}{2} \le a \le \frac{\sqrt{2\delta_0^2 + 2\sqrt{\delta_0^4 + 16c^2\tan\theta_0{}^2\delta_0^2}}}{4} \tag{4}$$

where c, σ_0, and δ_0 are input parameters. The lower bound of a corresponds to the existence of P_1 and P_2, which is consistent with previously reported works [14]. The upper bound corresponds to an ellipse with the minimal eccentricity for a given c value that satisfies the constraints, i.e. the widest possible ellipse for any given height. The lower bound of Eq. (4) is caused by the dependency between θ_0 and δ_0. Specifically, given the parameters c, θ_0, and δ_0, it is not possible to reduce a below this lower bound while satisfying the relevant constraints.

Notably, Eq. (4) has an important consequence. Given the height c of an ellipse, the size and shape of the ellipse is fixed. Hence, an ellipse equation can be determined from c, σ_0, and δ_0. In this paper, for any given c, an ellipse with the minimum possible eccentricity (i.e. the widest possible ellipse) is created as a corresponding hollow.

Eccentricity measures the shape of a conic curve, and plays an important role in this study as a control handle for ellipse and spheroid shape. The eccentricity ϵ of the ellipse in Eq. (2) is defined as:

$$\epsilon = \sqrt{1 - \frac{a^2}{c^2}}. \tag{5}$$

Hence, the eccentricity of a circle is 0, and that of an ellipse is less than 1. The range of ϵ of the ellipse corresponding to Eq. (4) is derived using Eq. (5) as follows:

$$\sqrt{1 - \frac{\delta_0^2 + \sqrt{\delta_0^4 + 16c^2\tan\theta_0{}^2\delta_0^2}}{8c^2}} \le \epsilon \le \sqrt{1 - \frac{\delta_0^2}{4c^2}} \tag{6}$$

The lower bound of Eq. (6) yields the widest ellipse possible. Given an eccentricity ϵ that satisfies Eq. (6), a can be determined from Eq. (5) as follows:

$$a = c\sqrt{1 - \epsilon^2}. \tag{7}$$

As aforementioned, this study employs prolate spheroids. Let S be a prolate spheroid and P be a plane passing through the major axis of S. Then, $\partial S \cap P$ is an ellipse and the ellipses corresponding to any P are congruent. Here we intentionally use eccentricity as a shape descriptor for spheroid S.

2.2. Hollowing with Spheroids using the Voronoi diagram of ellipsoids

Given a polyhedron $\mathcal{P}$, an optimal hollowing of $\mathcal{P}$, using a certain arrangement of spheroids, must be found. Specifically, we must determine the optimal arrangement of a set of m ellipsoids $\mathcal{E}3 = \{E3_1, E3_2, ..., E3_m\}$ within $\mathcal{P}$, to maximize the sum of all the ellipsoid volumes while satisfying the 3D printing process constraints. This is equivalent to minimizing the amount of material used to fill $\mathcal{P}$ by maximizing the volume of the ellipsoidal hollows that will be left unfilled. Notably, the increment of an ellipsoid terminates when one of the termination conditions is encountered, which may be prescribed by either the total volume or number of ellipsoids. Alternatively, the process may terminate when an ellipsoid increment cannot exist without violating the prescribed minimum wall thickness. Overall, the ellipsoid packing problem is not any easier than the disk packing problem, which is NP-hard [27,28]. The algorithm used here is based on Voronoi diagrams, which are powerful tools for spatial reasoning among geometric objects. A brief review of the various Voronoi diagrams used in this paper is provided in Appendix C. Additionally, we present this problem in a general setting of ellipsoids, rather than the special case of spheroids.

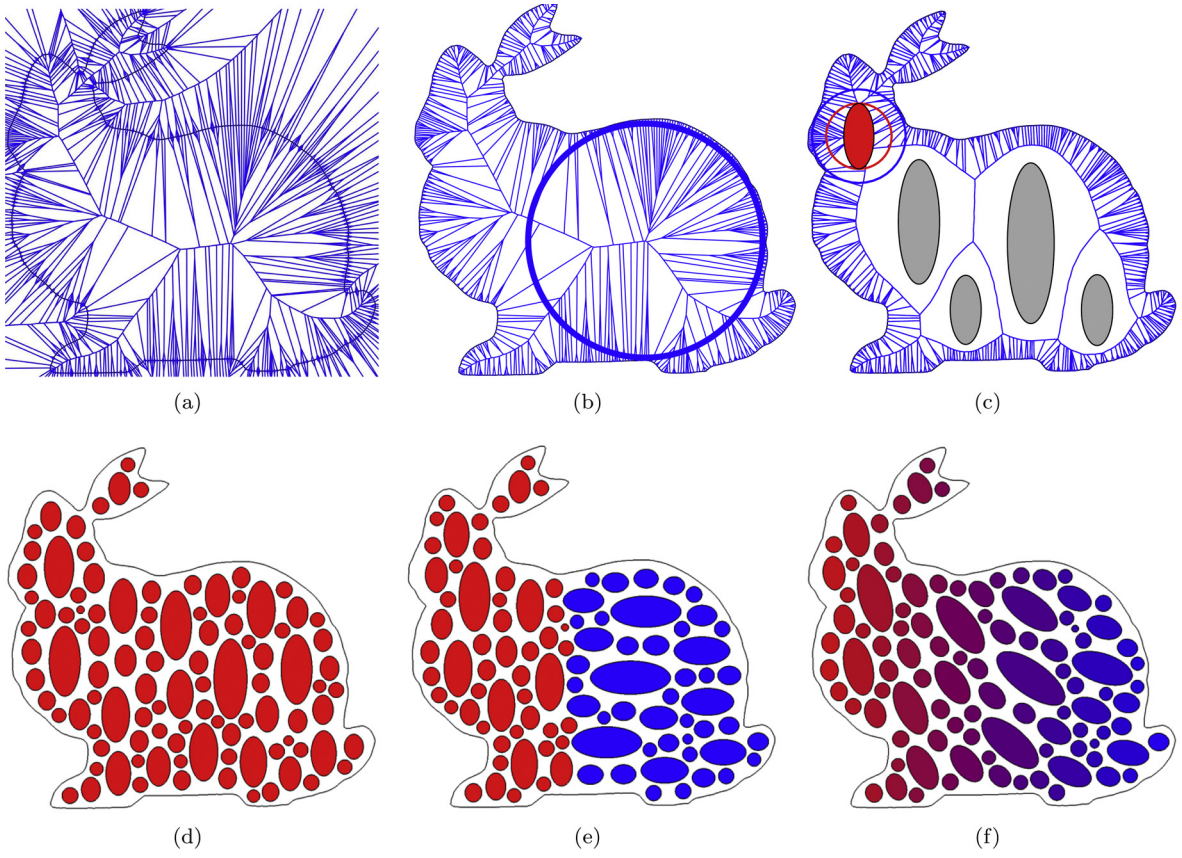


Fig. 2. The ellipse arrangement in a bunny polygon (758 edges on the boundary). (a) The entire Voronoi diagram of the bunny ($\widehat{\text{VD}}$(Poly2)). (b) The interior subset of the Voronoi diagram (VD(Poly2)). The blue circle represents the maximum clearance probe of the bunny. (c) The fifth ellipse to be placed within the red ellipse mask after four other ellipses were incremented ($\text{VD}_4 = \text{VD}(\mathcal{E}2_4, \text{Poly2})$, where $\mathcal{E}2_4 = \{E2_1, E2_2, E2_3, E2_4\}$). The red circle represents an ellipse mask that was shrunken from the blue circle. (d), (e), and (f) Ellipses placed according to different practical criteria. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2.1. Arranging ellipses in a 2D polygon

A brief review is provided here for the ellipse packing algorithm in a 2D polygon, Poly2, with further details provided in [14]. The idea is similar to that for its 3D counterpart. Fig. 2(a) shows the Voronoi diagram $\widehat{\text{VD}}$(Poly2) of a simple polygon, Poly2, which is obtained from the intersection between the Stanford bunny and a section plane. The bunny polygon Poly2 has 758 oriented line segments. $\widehat{\text{VD}}$(Poly2) consists of two subsets, the interior and exterior, with respect to the boundary of Poly2. Let VD(Poly2) be the interior subset, as shown in Fig. 2(b).

In VD(Poly2), each V-vertex v is associated with a maximum empty circle that touches three generators, called the **clearance probe** π_v, centered at v. The radius of π_v is the distance from v to its generators. Notably, different V-vertices may have varying clearance values. The **maximum clearance probe** $\pi_{\max}$ is the largest clearance probe that can be defined at a V-vertex $v_{\max}$, and can be found in $O(n)$ time from VD (Poly2) of n generator entities. The blue circle in Fig. 2(b) is the first maximum clearance probe of the bunny.

The Voronoi diagram VD($\mathcal{E}2$,Poly2) of a set $\mathcal{E}2$ of ellipses within Poly2 is constructed as follows. Consider VD(Poly2) as the initial Voronoi diagram $\text{VD}_0 = \text{VD}(\mathcal{E}2_0, \text{Poly2})$, where $\mathcal{E}2_0 = \{\varnothing\}$. Starting from VD_0, the algorithm first finds the V-vertex $v_{\max}$ with $\pi_{\max}$ of VD_0. Then, it places an ellipse $E2_1 \subseteq \pi_{\max}$, centered at $v_{\max}$, and updates the Voronoi diagram to obtain $\text{VD}_1 = \text{VD}(\mathcal{E}2_1, \text{Poly2})$, where $\mathcal{E}2_1 = \{E2_1\}$. The algorithm repeats the two steps described above as many times as it requires. As shown in Fig. 2(c), VD_4 and the maximum clearance probe has a radius $r_{\max}$, shown as the blue circle, which touches exactly three points on the boundary of the bunny. Any ellipse contained within this probe is free from intersecting with any other entities of the model. To guarantee that the prescribed minimum wall thickness τ_{btw} and τ_{bndry} is maintained, the probe shrinks with a **shrinkage** ξ to reach the red circle, called an **ellipse mask** μ with the radius $r_\mu (=\xi r_{\max})$, $0 \le \xi \le 1$. In 3D, μ is called an **ellipsoid mask**. The purpose of ξ is to provide the user a convenient handle for controlling the distribution of ellipse heights. The ellipse mask in Fig. 2(c) is defined by $\xi = 0.7$, and the red ellipse inscribes the mask. Fig. 2(d)–(f) illustrate arrangements of 100 ellipses each, which are defined according to specific application needs. In Fig. 2(d), the major axes of the ellipses are all vertically-aligned. In Fig. 2(e), the right half of each ellipse is aligned horizontally. In Fig. 2(f), the ellipse orientations are determined as a function of their horizontal locations. The transition of the ellipse colors, which correspond with orientation is noticeably smooth.

The key, challenging step in the algorithm is its insertion of each ellipse into an existing Voronoi diagram. Given $\text{VD}_i = \text{VD}(\mathcal{E}2_i, \text{Poly2})$, where $\mathcal{E}2_i = \{\text{E2}_1, \text{E2}_2, \ldots, \text{E2}_i\}$, the topology-oriented incremental algorithm increments E2_{i+1} into VD_i to get VD_{i+1}. Because there is no known analytical method to correctly and efficiently find V-vertices in a computationally tractable manner, a numerical method is used to solve this problem, as follows. Further details can be found in [14]. Let C2 be a set of circles inscribing E2_{i+1}. Hence, C2 approximates E2_{i+1}. The algorithm first individually increments each circle in C2 into VD_i, using the topology-oriented incremental algorithm in [29]. After this, it merges the V-cells of the circles in C2, producing an accurate topology structure of VD_{i+1}. The algorithm then computes the coordinates of each new V-vertex by a numerical iteration, which converges quickly.

2.2.2. The arrangement of spheroids in a polyhedron in 3D

The use of the algorithm to find an ellipsoid arrangement, specifically for the vertically-aligned prolate spheroids in this paper, in a polyhedral CAD model $\mathcal{P}$(≡Poly3) in 3D is conceptually similar to that of the ellipses in 2D described previously. Thus, the idea itself is simple, but constructing it in an accurate and efficient manner has been a significant challenge, and to the best of our knowledge this has not been accomplished previously.

The algorithm consists of two major steps: (i) constructing the Voronoi diagram of the polyhedron $\mathcal{P}$, and (ii) inserting a new ellipsoid into the current Voronoi diagram of ellipsoids. Both steps rely heavily on the topology-oriented incremental construction of 3D spherical balls, which was recently reported in [30,31]. The overview of the algorithm is presented here, with the details forthcoming in another mathematical paper.

Constructing the initial Voronoi polyhedron. The Voronoi diagram $\mathcal{VD}(\mathcal{P})$ of a polyhedron $\mathcal{P}$ is a spatial tessellation, such that every point in the space is assigned to the closest entity, either a vertex, edge, or triangular face, of $\mathcal{P}$. The Voronoi diagram $\mathcal{VD}(\mathcal{P})$ is represented by a quadruplet (V, E, F, C), which represent the sets of V-vertices, V-edges, V-faces, and V-cells, respectively. The correct and efficient construction of $\mathcal{VD}(\mathcal{P})$ still posed a challenge.

Generally, the correct topology of the Voronoi diagram and the correct coordinate of its V-vertices are sufficient to arrange ellipsoids in a polyhedron, and most applications, including the current one, do not require the geometry of V-edges and V-faces. In the rare case where it is required, this geometry can be computed either analytically or numerically, depending on the combination of generating entities for a given topology. To obtain the correct topology, the topology-oriented incremental construction approach is commonly used [32,29,30].

A polyhedron $\mathcal{P}$ is represented as a triplet ($V^{\mathcal{P}}$, $E^{\mathcal{P}}$, $F^{\mathcal{P}}$), where $V^{\mathcal{P}}$, $E^{\mathcal{P}}$, and $F^{\mathcal{P}}$ represent the sets of vertices, edges, and faces of $\mathcal{P}$, respectively. Consider the ordinary Voronoi diagram $\widehat{\mathrm{VD}}(V^{\mathcal{P}})$ of the vertices. If $V^{\mathcal{P}}$ contains a sufficient number of vertices from the boundary of $\mathcal{P}$, the structure of $\widehat{\mathrm{VD}}(V^{\mathcal{P}})$ converges to that of the Voronoi diagram $\mathcal{VD}(\mathcal{P})$ of $\mathcal{P}$, which can be used to correctly place ellipsoids. The approximation of $\mathcal{VD}(\mathcal{P})$ is then constructed by the following algorithm.

First, $\widehat{\mathrm{VD}}(V^{\mathcal{P}})$ is constructed using the topology-oriented incremental algorithm [30,31] to obtain the representation (V^V, E^V, F^V, C^V) of the sets of V-vertices, V-edges, V-faces, and V-cells, respectively. V^V is then classified into two subsets, the interior subset V^V_{Int} and the exterior subset V^V_{Ext}, with respect to the $\mathcal{P}$ boundary. E^V, F^V, and C^V are similarly classified. Let VD = VD(V^V_{Int}, E^V_{Int}, F^V_{Int}, C^V_{Int}) be the Voronoi diagram in $\mathcal{P}$. Then, VD approximates $\mathcal{VD}(\mathcal{E}3, \mathcal{P})$, where $\mathcal{E}3 = \{\emptyset\}$.

Locating a new ellipsoid. Given $\mathcal{VD}(\mathcal{E}3, \mathcal{P})$, the V-vertex $v_{\max} \in V^V_{\mathrm{Int}}$ that corresponds to the maximum clearance probe $\pi_{\max}$ can be easily found, and the ellipsoid mask μ can be easily computed, by applying the shrinkage ξ to $\pi_{\max}$. Then, a new ellipsoid is defined at $v_{\max}$, and the Voronoi diagram is updated within $\mathcal{P}$. During each ellipsoid increment, it is necessary to check whether the minimum wall thicknesses τ_{bndry} and τ_{btw} conditions are successfully satisfied.

Updating the Voronoi diagram with an incremented ellipsoid. The idea for constructing the Voronoi diagram $\mathcal{VD}(\mathcal{E}3, \mathcal{P})$ of a polyhedron $\mathcal{P}$ and an ellipsoid set $\mathcal{E}3$ is conceptually similar to in the process followed for a 2D scenario in Section 2.2.1. Notably, it is actually very difficult to accomplish this construction in 3D correctly and efficiently. Thus, the algorithm details are beyond the scope of this paper and will instead be reported in another forthcoming paper, but a brief overview is provided here. An ellipsoid E3 is approximated by a set B3 of inscribing balls, thus called **inballs**, which can be produced in a variety of ways. An inball may intersect another, but cannot fully contain another. Given $\mathcal{VD}_i = \mathcal{VD}(\mathcal{E}3_i, \mathcal{P})$, each of the inballs in B3 is incrementally inserted, and the associated V-cells are merged to obtain the $\mathcal{VD}_{i+1}$ topology. Fig. 3 shows some of the details involved in the incremental process of constructing spheroids in the cube: (a) the boundary of cube $\mathcal{P}$ after being triangulated a few times, (b) the interior subset of the approximated Voronoi diagram of $\mathcal{P}$, (c) a spheroid incremented in the approximated Voronoi diagram, and (d) the approximated Voronoi diagram, after five spheroids have been incremented. Furthermore, Fig. 4 shows the spheroid increment process for bunny and dog models. Notably, each large planar face is triangulated with multiple triangular faces, because the construction of the correct Voronoi diagram of a polyhedron with ellipsoidal hollows has not yet been developed.

Solution accuracy. The accuracy of the approximation VD for the Voronoi diagram $\mathcal{VD}(\mathcal{P})$ is critical for ellipsoids to be placed accurately in $\mathcal{P}$ and depends directly on the size of $V^{\mathcal{P}}$. Specifically, a larger $|V^{\mathcal{P}}|$ corresponds to a more accurate VD. Hence, it may be necessary or desirable to subdivide each face $f \in F$ into multiple triangles to achieve an acceptable accuracy level. Four different instances of refinements made to the cube model and the corresponding Voronoi diagrams are shown in Fig. 5.

Computational efficiency. Fig. 6 shows the computation time profile of the proposed algorithm for arranging the spheroids. The black curve at the top corresponds to the cube model shown in Fig. 3, where every planar face is triangulated by multiple triangular faces. The blue and red curves correspond to the bunny and dog models, respectively, shown in Fig. 4. The horizontal and vertical axes denote the number of spheroids and computation time, respectively. Clearly, the computation time is linear to the number of spheroids.

2.3. Generating tool paths using G2/G3-codes

The 3D printer used in this work is an NC machine. To print an artifact, a sequence of G-codes had to be generated to drive the extruder for each printing layer. There are several popular software tools for this purpose, such as Slic3r, an open source freeware [33] and Ultimaker Cura, a freeware [34].

2.3.1. Advantages of using G2/G3-codes over G1-codes

One common approach for driving an extruder is to use linear motions based on G1-codes. Circular motions, however, offer several advantages over linear motions: (i) shorter printing time, (ii) better printing quality around curved boundaries, and (iii) smaller tool path file size.

Suppose that a circle C, or an arc on C, is intended to be printed, and that C is approximated by a set of line segments, each of which will be printed by a linear motion using a G1-code. To make sure the line segments satisfy a priori defined quality constraint, a large number of line segments, each corresponding to a G1-code, must be used. However, if a circular interpolator is available, a single G2/G3-code can be used to print C in its entirety.

A tool motion consists of three sub-segments: acceleration, deceleration, and moving at a constant speed, called a **feed rate**. Hence, if C is printed by a large number of G1-codes, there are an equally large number of acceleration and deceleration pairs. On the other hand, if a G2 or G3-code is used, C can be printed with just one angular acceleration, one angular deceleration, and one segment of constant angular velocity. Consequently, printing C using a single circular motion is much faster than printing it with many linear motions.

Another critical issue is the printing quality, which must be addressed from two perspectives: (i) the instant printing pause between two consecutive motions, and (ii) the approximation error of linear motions for printing curved features. First, consider the linear motions for two consecutive line segments L_i and L_{i+1}, where $L_i \cap L_{i+1} = p$. Both an overshooting and a short moment with no motion at p exist because of the deceleration for L_i and the acceleration for L_{i+1}. This causes an uneven extrusion of material, as the material extrusion rate is fixed and cannot be adjusted to accommodate this temporary stay. Thus, the printing quality suffers accordingly, and is lower than that obtained

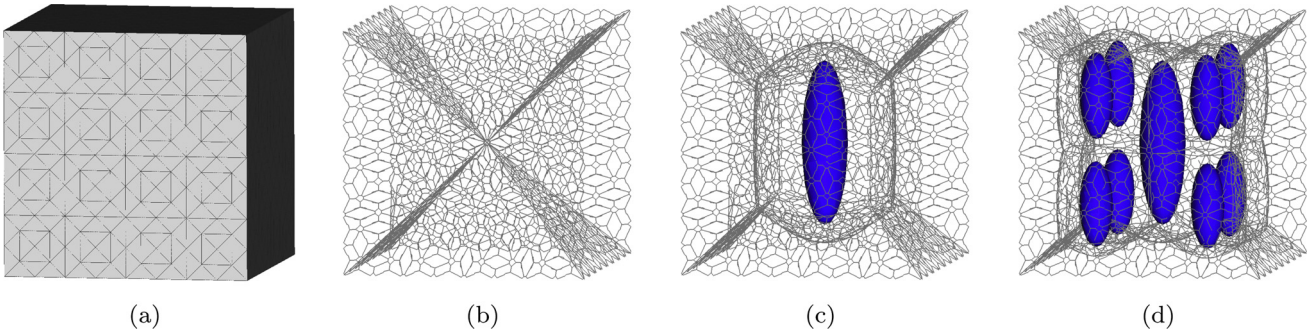


Fig. 3. Incrementing spheroids into a cube using the Voronoi diagram of a cube with ellipsoids. (a) A cube $\mathcal{P}$ with triangulated faces represented as a triplet ($V^{\mathcal{P}}$, $E^{\mathcal{P}}$, $F^{\mathcal{P}}$) of the sets of vertices, edges, and faces ($|V^{\mathcal{P}}|$: 770, $|E^{\mathcal{P}}|$: 2304, $|F^{\mathcal{P}}|$: 1536). (b) The interior subset of the ordinary Voronoi diagram VD(V) of V, which approximates the interior of the Voronoi diagram of $\mathcal{P}$. (c) The Voronoi diagram $\mathcal{VD}(\mathcal{E}3 = \{E3_1\}, \mathcal{P})$ after the first ellipsoid $E3_1$ is incremented. (d) The Voronoi diagram $\mathcal{VD}(\mathcal{E}3 = \{E3_1, E3_2...,E3_5\}, \mathcal{P})$ after the fifth ellipsoid $E3_5$ is incremented.

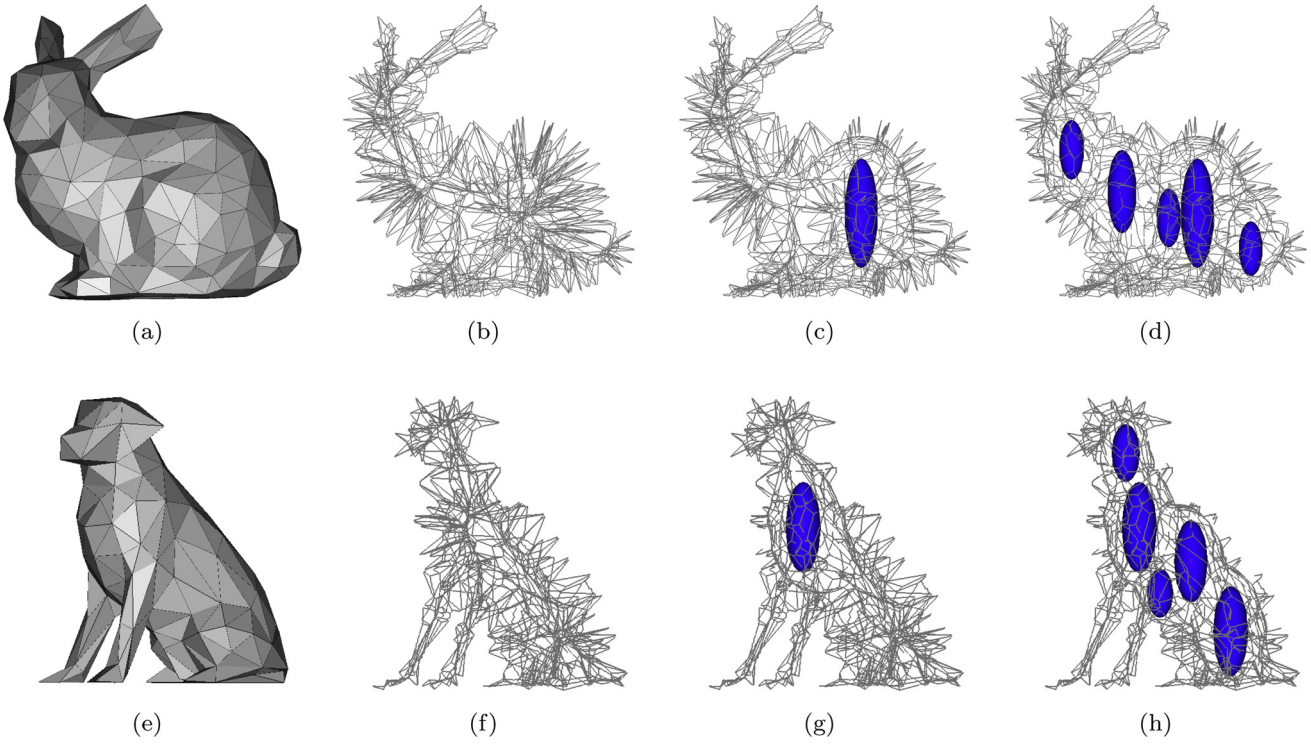


Fig. 4. Incrementing spheroids in the bunny (with 500 faces) and dog (with 300 faces) models using the Voronoi diagram of polyhedron with ellipsoids. (a) The bunny model $\mathcal{P}$ ($|V^{\mathcal{P}}|$: 252, $|E^{\mathcal{P}}|$: 750, $|F^{\mathcal{P}}|$: 500). (b) The interior subset of the ordinary Voronoi diagram VD(V) of V, which approximates the interior of the Voronoi diagram of $\mathcal{P}$. (c) The Voronoi diagram $\mathcal{VD}(\mathcal{E}3 = \{E3_1\}, \mathcal{P})$ after the first ellipsoid $E3_1$ has been incremented. (d) The Voronoi diagram $\mathcal{VD}(\mathcal{E}3 = \{E3_1, E3_2...,E3_5\}, \mathcal{P})$ after the fifth ellipsoid $E3_5$ has been incremented. (e–h) The same spheroid increment process applied to the dog model $\mathcal{P}$ ($|V^{\mathcal{P}}|$: 152, $|E^{\mathcal{P}}|$: 450, $|F^{\mathcal{P}}|$: 300). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using a single command for circular interpolator. Fig. 7 shows an extreme case of this, demonstrating how the printing quality is affected by two consecutive linear motions. The tool path, consisting of G1-codes to print a planar, rectangular surface, is shown in Fig. 7(a). Note that there are long linear motions at the corners of the square. The resulting printed artifact is shown in Fig. 7(b) and (c), and the overshooting at the four corners can be clearly observed in both figures, consistent with other recent reports [35]. Notably, this can happen at any vertex between two consecutive G1-codes. Additionally, the linear approximation of a circular feature produces an approximation error, termed α_1, which depends on the path generation software used. If a user is equipped with G2/G3-codes, the error $\alpha_2 << \alpha_1$ is then determined by the circular motion interpolator design, which can in most cases be equivalent to the machine resolution of the specific printer used.

2.3.2. Spheroidal hollows and G2/G3-codes via offsetting

Let O be the CAD model of an object with prolate spheroidal hollows, whose major axes are parallel to the Z-axis. This is the artifact to be printed. Let Poly2 be the layer to be printed, which is defined by the intersection of O with a section a plane P parallel to the XY-plane. Then, Poly2 is a polygon containing non-intersecting circles, where each circle corresponds to a spheroid.

Fig. 8(a) shows a rectangular polygon Poly2 = {Bndry, D1, D2, D3, D4}, which contains four circular disks and the Voronoi diagram VD (Poly2). Let ω be the width between two consecutive offsets, or between the first offset and the polygon boundary, and let Ω be the width between the offset and the polygon boundary. Fig. 8(b) shows the (red) offset of Poly2, where the offset amount $\Omega = \omega$ for a small value of ω. In this case, there are no offsets of the circles or rectangle intersections. When $\Omega = 2\omega$ (Fig. 8(c)), the offset of Bndry intersects the offsets of both D_1 and D_3. Furthermore, the offsets of D_2 and D_4 intersect as well.

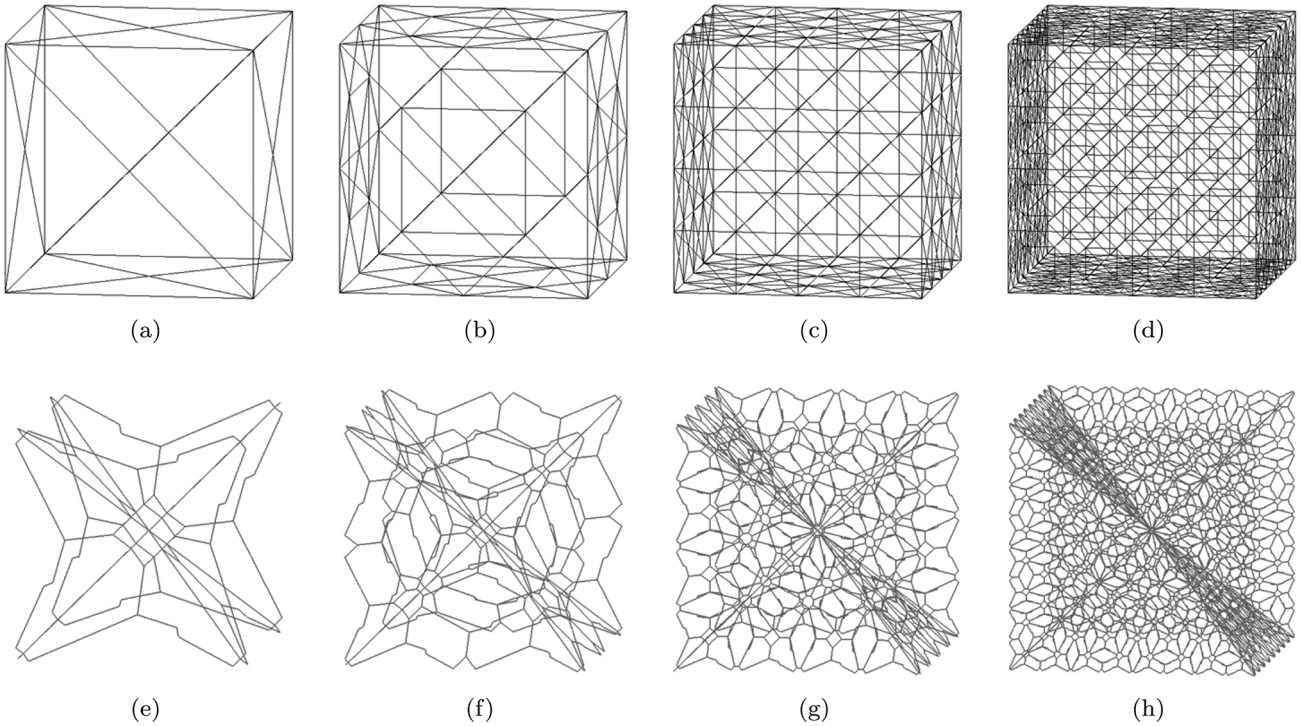


Fig. 5. Varying cube model resolution and the corresponding Voronoi diagrams. (a–d) cube models, (e–h) corresponding Voronoi diagrams. (a) and (e) The lowest resolution. ($|V^{\mathcal{P}}|$: 14, $|E^{\mathcal{P}}|$: 36, $|F^{\mathcal{P}}|$: 24). (b) and (f) Low resolution, after each face of (a) is subdivided once. ($|V^{\mathcal{P}}|$: 50, $|E^{\mathcal{P}}|$: 144, $|F^{\mathcal{P}}|$: 96). (c) and (g) Intermediate resolution, after each face of (b) is subdivided once. ($|V^{\mathcal{P}}|$: 194, $|E^{\mathcal{P}}|$: 576, $|F^{\mathcal{P}}|$: 384). (d) and (h) The highest resolution after each face of (c) is subdivided once. ($|V^{\mathcal{P}}|$: 770, $|E^{\mathcal{P}}|$: 2304, $|F^{\mathcal{P}}|$: 1536).

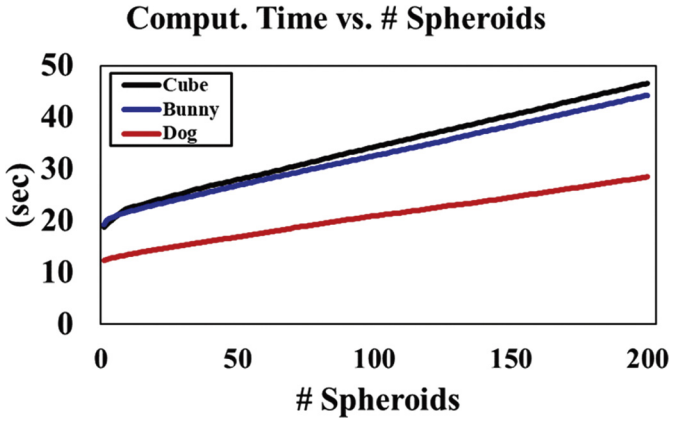


Fig. 6. Computation time profile for arranging spheroids in an object represented by a refined triangular mesh. Black curve: Cube ($|V^{\mathcal{P}}|$: 6146) Blue curve: Bunny ($|V^{\mathcal{P}}|$: 5374) Red curve: Dog ($|V^{\mathcal{P}}|$: 3633). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

When $\Omega = 3\omega$ (Fig. 8(d)), he offset of Bndry intersects the offsets of D_1 and D_2, the offsets of D_2 and D_3, the offsets of D_2 and D_4, etc. Additionally, the offsets of D_1, D_2, and D_4 intersect in this case.

In Fig. 8(d) we observe that the intersections between two consecutive contour-parallel offsets occur on the blue edges of the Voronoi diagram. Hence, if VD(Poly2) is constructed as a preprocessing step, the offset computation can be facilitated. In this study, this was done using the algorithm reported in [14]. Then, each offset path for $\Omega = i\omega$ for $i = 1, 2, \ldots$ is generated in $O(m)$ time, using a modified version of the algorithm reported in [36,37], where m represents the number of entities in Poly2. The offset generation algorithm is very simple and proceeds as follows. It first scans the Voronoi vertices to calculate the clearance of every vertex and mark those with a clearance greater than Ω as "active". Then, it finds the subgraph of VD(Poly2). Starting from an active vertex, it traverses all incident edges where an edge e intersects an offset of a related entity. After the propagation is completed, the offset chain is generated by appropriately connecting the offset vertices.

Fig. 8(e) shows Poly2, together with its Voronoi diagram VD(Poly2), where Poly2 is a square 100×100 mm with 21 mutually exclusive disks. The contour-parallel offset produced, using the Voronoi diagram with the proposed algorithm, with $\omega = 2$ mm for visualization, is shown in Fig. 8(f). Importantly, the Voronoi edges are not part of tool paths.

3. Results and discussion

The primary products of this study are: (i) the support-free hollowing algorithm based on prolate spheroids, (ii) the tool path generation algorithm relying heavily on circular printing motions with G2/G3-codes, and (iii) the HollowTron webserver, which could implement the two algorithms.

HollowTron takes an input of a 3D CAD solid model, represented as a triangular mesh stored in an STL file, and arranges spheroidal hollows within the boundary of the CAD model (Fig. 9). The geometric model, with these generated spheroids, is stored in "Hollowed Model". There are two important outputs. For engineering analysis using ANSYS, "Mesh Model" is stored in the STL format after the spheroids are triangulated. For the printing process, a tool path file, consisting of G1, G2, and G3-codes, is stored in a "Tool Path" file. Fig. 9 illustrates the relationship among HollowTron, the ANSYS analysis, and a commercial 3D printer. Thorough experiments were performed with the following settings.

Computational platform. The HollowTron webserver embeds the APIs of the V library, developed by Voronoi Diagram Research Center using C++. Stress and deformation analyses were conducted on a desktop computer, specifically an Intel(R) Core(Tm) i7-8700 CPU@ 3.2 GHz with 16 GB RAM.

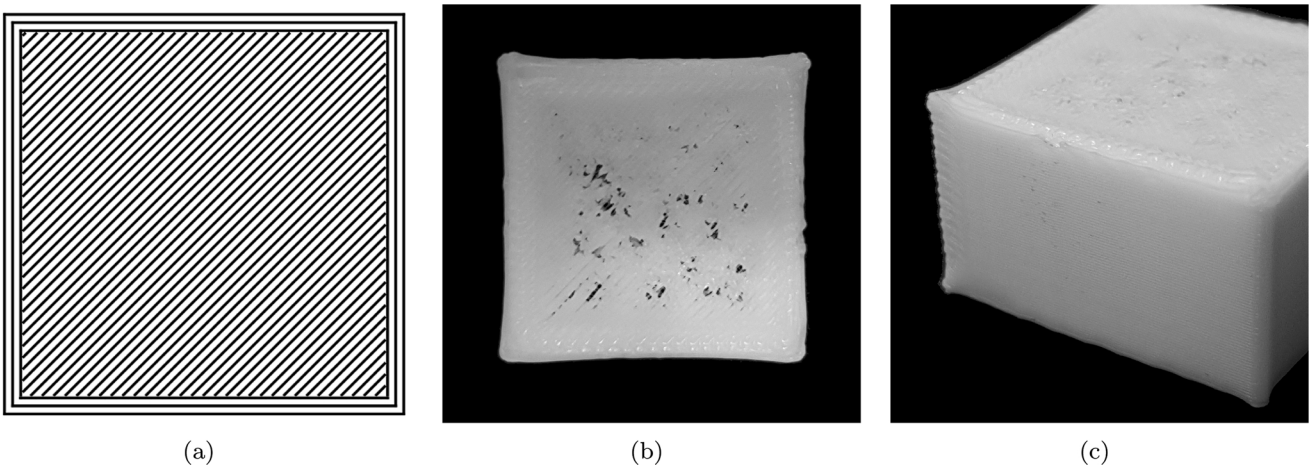


Fig. 7. Illustration demonstrating the overshooting at the sharp corners. (a) Printing tool path for each layer. (b) Top view of the printed artifact. (c) Another view of the printed artifact.

3D printer. Artifacts were printed on a commercial extrusion-based 3D printer SINDOH 3DWOX 1X. The tray size was 228 mm × 200 mm × 300 mm, the default printing bed temperature was 60 °C, nozzle temperature was 200 °C, nozzle printing speed was 40 mm/s and was adjustable between 10 mm/s and 200 mm/s), the minimum wall thicknesses was τ_{btw} 1.0 mm between two hollows and τ_{bndry} 1.0 mm between the hollows and model boundaries. As an example of the types of structures used in this work, Fig. 10 shows a bunny printed on top of a 2 mm raft with an external support structure. In this study, it is important to distinguish that "support-free" refers

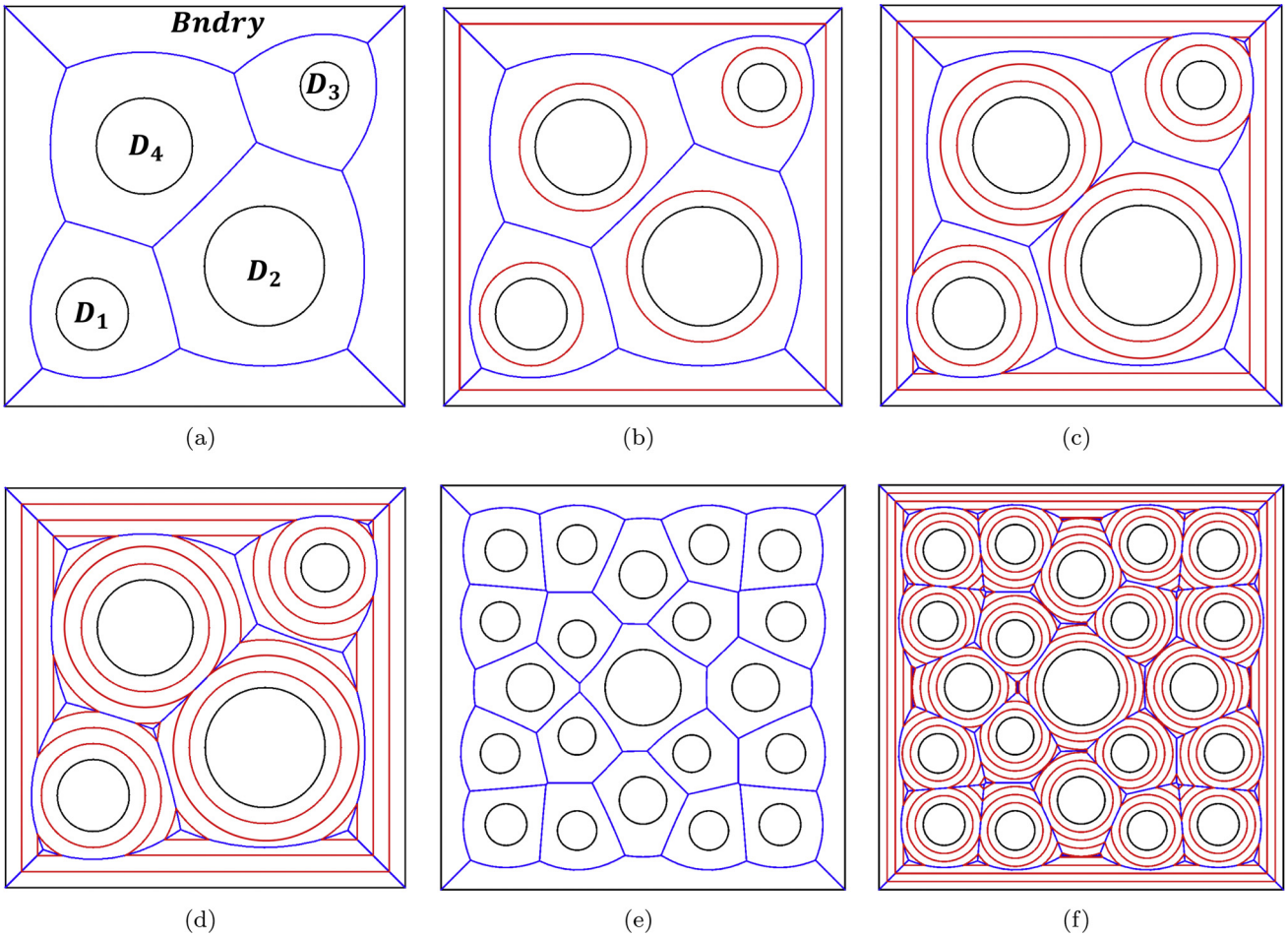


Fig. 8. Contour-parallel offset generation of a polygon Poly2 using its Voronoi diagram VD(Poly2). Black: Input polygon and circular disks within. Blue: The Voronoi diagram VD(Poly2). Red: The contour-parallel offsets. (a) Poly2, which is {Bndry, D_1, D_2, D_3, D_4}, and its Voronoi diagram VD(Poly2). (b) The offset of Poly2 corresponding to $\Omega = \omega > 0$. No offset intersects any other. (c) The offset corresponding to $\Omega = 2\omega$. (d) The offset corresponding to $\Omega = 3\omega$. (e) A polygon Poly2 (in black) with 21 circular disks and its Voronoi diagram VD(Poly2) (in blue). (f) Contour-parallel offsets (in red) computed using the Voronoi diagram and the algorithm in [36,37]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

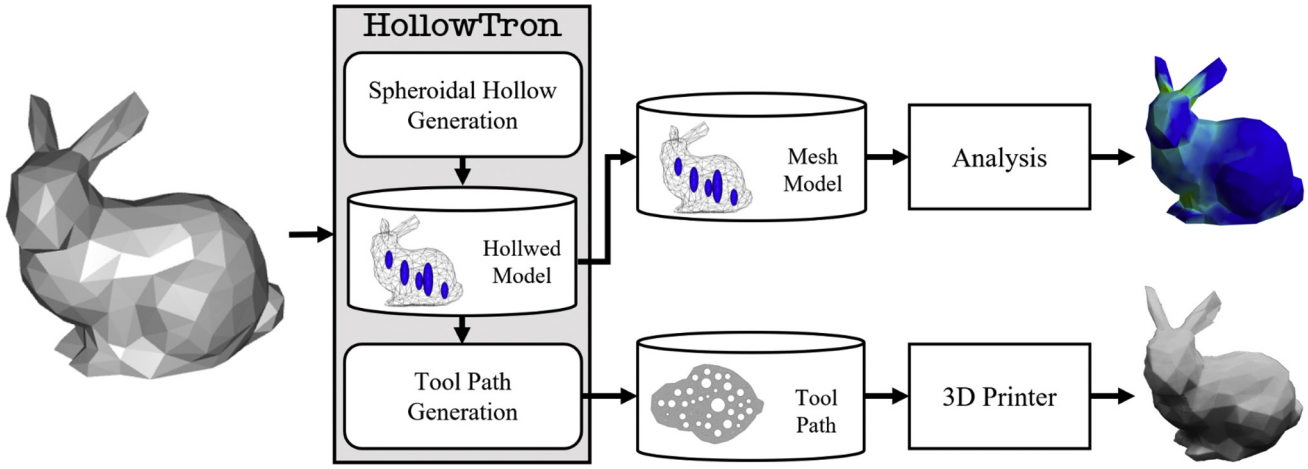


Fig. 9. The relationship among HollowTron, the ANSYS analysis, and a 3D printer.

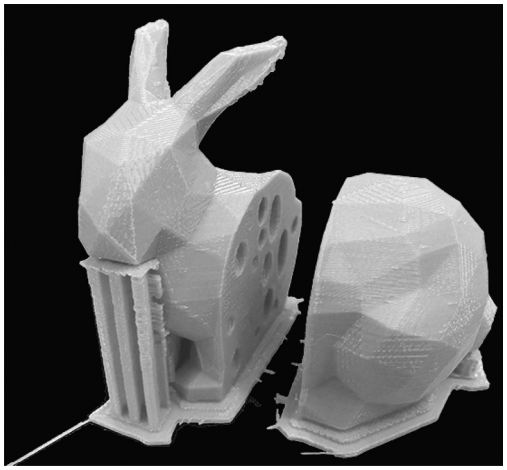


Fig. 10. The bunny model printed with an external support structure on a raft.

only to the interior hollows, and exterior support structures may exist.

Material specification. Plastic PLA (polylactic acid) was used as the material to print artifacts with the following parameters. The maximum allowed overhang angle (θ_0) was 60°, the maximum overhang length was (δ_0) 5 mm, and the printing layer thickness (σ_0) was 0.2 mm.

Manufacturing setting. In general, all mentioned CAD models containing spheroidal hollows were printed. To verify that the overhangs of support-free hollows would not fall, we subdivided and printed a few models through some section planes, so that hollow interior could be visually checked. However, exterior support structures were generated wherever necessary, and were removed after printing was completed.

3.1. Hollowed CAD models and analyses

Given an object CAD model, Fig. 9 shows an arrangement of spheroids within the object boundary, determined using HollowTron. To thoroughly compare our proposed method with other methods, we performed both stress and deformation analyses using ANSYS Structure [38]. Three-dimensional tetrahedral meshes were generated by ANSYS Space Claim [39], and PLA was modeled as an isotropic linear elastic. For the FEM analysis, the following parameters were used: A density of 1,240 kg/m^3, Young's Modulus of 3.5 GPa, and Poisson's ratio of 0.36.

First, we performed a compression test by uniformly applying 5 MPa pressure to the top of a 100 mm × 100 mm × 100 mm cube model. The cube models with hollows defined by an octahedron and an ellipsoid are illustrated in Fig. 11(a) and (d), respectively, where both hollow shapes occupy 3.6% of the total cube volume. The von Mises stress distributions, viewed from the front, are presented in Fig. 11(b) and (e), and Fig. 11(b) clearly shows the high-stress region on the sharp edges of the octahedron, which agrees with our initial expectation. Importantly, the highest stress observed in the ellipsoidal hollow in Fig. 11(e) is significantly lower than that in the octahedral hollow in Fig. 11(b). The deformation analysis results are shown in Fig. 11(c) and (f), and indicate that the ellipsoidal hollow deforms less severely.

The von Mises stress analysis results for three different hollow methods, applied to the same cube, are shown in Fig. 12. In all cases, forces are applied uniformly to the cube tops. Fig. 12(a) shows a large rhombic hole, with a few additional smaller rhombic holes positioned nearby. Notably, all the holes are located fully inside the cube and do not intersect each other, maintaining a minimum distance of 1.5 mm between adjacent rhombuses. Let **infill density** ρ be defined as:

$$\rho = \frac{\text{Vol(Poly3)} - \text{Vol(Hollows)}}{\text{Vol(Poly3)}} \tag{8}$$

Where Vol(Poly3) is the total volume of the polyhedron Poly3 and Vol (Hollows) is the total volume of all hollows within Poly3. In Fig. 12(a), $\rho = 0.52$, and a sectional view of the hollow arrangement is provided. Fig. 12(e) and (i) demonstrate arrangements of congruent rhombic holes, with $\rho = 0.52$, and elliptic cylinders, with $\rho = 0.53$, respectively. The latter was created by the algorithm in [14]. The second and third columns in the figure illustrate the isometric and sectional views, respectively, of the von Mises stress analysis results for the corresponding models. The fourth column shows the total deformation analysis results. The legends in the vertical bars to the right of the third and fourth columns cover an identical value range.

In Fig. 12(b) and (c), a high stress concentration is observed near both the cube wall and the ends of the rhombic hollows. This phenomenon is exacerbated for rhombic hollows located near the top of the cube, which are physically closer to the force application. Fig. 12(f) and (g) exhibit similar stress concentrations, but the stress is more widely distributed among the rhombic hollows. For the structures in Fig. 12(j) and (k), the elliptic cylindrical hollows experience stress that is more uniformly and widely distributed than the two rhombic methods. Fig. 12(d), (h), and (l) show the total deformation analysis results for the corresponding cases. In Fig. 12(d) and (h), a serious deformation, covering most of the upper region of the cube, is observed. Contrarily, significantly lower deformation, which is distributed evenly throughout the cube, is shown in Fig. 12(l). Overall, these simulation results clearly indicate that the elliptic cylindrical hollow is superior to the other evaluated methods.

The proposed ellipsoidal hollow method was used to create several models with HollowTron, and the same uniform compression test was

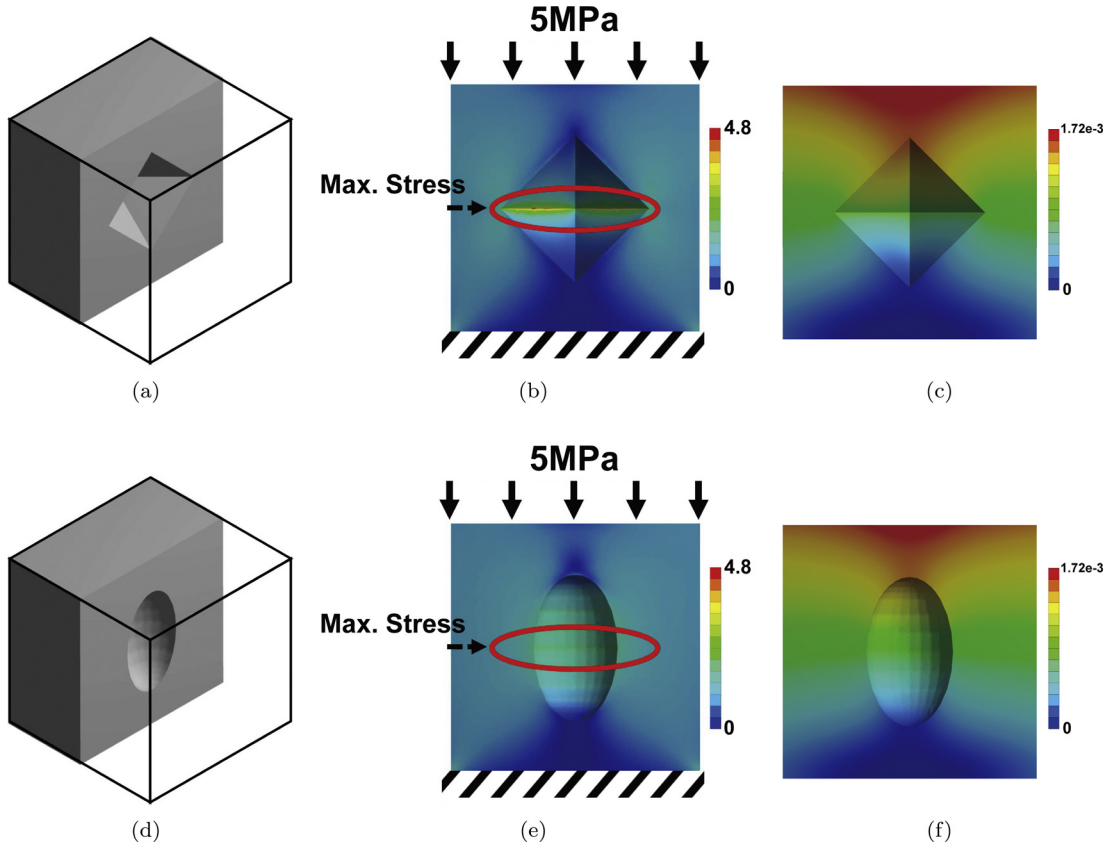


Fig. 11. The von Mises stress and total deformation of cube models with spheroidal and octahedral hollows. Pressure was applied uniformly to the cube tops, as represented by the arrows. The dimension of the cube was 100 mm × 100 mm × 100 mm. (a–c) Octahedral hollow results. (d–f) Spheroidal hollow results. The volumes of both octahedron and ellipsoid hollows are 3.6% of the total cube volume. 1st column: CAD model. 2nd column: von Mises stress distribution, with the results normalized by the applied 5 MPa pressure. 3rd column: Total deformation, with the results normalized by the height of the cube, 100 mm.

performed, as previously described. Fig. 13 presents nine cube models with spheroidal hollows generated using HollowTron with varied parameters, including shrinkage and infill density. Fig. 13(a) shows the CAD model with spheroids generated with a shrinkage $\xi = 0.9$ and infill density $\rho = 0.9$, denoted as CUBE($\xi = 0.9$, $\rho = 0.9$). The other cube models included in Fig. 13 are denoted as follows: (b) CUBE(0.9, 0.8), (c) CUBE(0.9, 0.7), (d) CUBE(0.8, 0.9), (e) CUBE(0.8, 0.8), (f) CUBE(0.8, 0.7), (g) CUBE(0.7, 0.9), (h) CUBE(0.7, 0.8), and (i) CUBE(0.7, 0.7). The corresponding von Mises stress analysis results for these models are provided in Fig. 14, visualized at the middle section plane. Table 1 summarizes the simulation results presented in Fig. 14, specifying the maximum von Mises stresses and total deformations. Ultimately, this numerical simulation confirms that support-free hollowing, based on vertically aligned prolate spheroids, exhibits a significantly low stress concentration and endures compressive strain under a more constant stress than other evaluated methods are capable of.

3.2. Printing artifacts: benchmark for tool path generation methods

To validate and verify the proposed method, several CAD models with various interior hollows were printed. Fig. 15 provides an overview of all the printed cubes, indicating the different types of hollows used. Each column in the figure corresponds to a different hollow type. The 1st column (a and e) shows elliptic cylindrical hollows ($\xi = 0.9$, $\rho = 0.5$), while the remaining columns show spheroidal hollows: the 2nd column (b and f) shows CUBE(0.9, 0.9), the 3rd column (c and g) shows CUBE(0.9, 0.8), and the 4th column (d and h)shows CUBE(0.9, 0.7). The cubes were trimmed using planar facets to reveal the interior structures. Furthermore, Fig. 16 presents two additional cube models, with the same infill ratio, $\rho = 0.7$, but different shrinkage values, ξ= 0.7 and 0.8. These can be directly compared with CUBE(0.9, 0.7) in Fig. 15(d) and (h). Additionally, Fig. 17(a) and (b) illustrate the printed bunny model BUNNY($\xi = 0.9$, $\rho = 0.8$), while the printed dog model DOG($\xi = 0.9$, $\rho = 0.8$) is shown in Fig. 17(c) and (d). To visualize the structure interiors, each model was subdivided into four pieces via three section planes.

Fig. 18 shows the various methods used to print one layer of the Poly2 data in Fig. 8(e). The layer to be printed is produced by a section plane P, which cuts through CUBE(0.7, 0.9) in the middle, so the equation of P is $Z = 50$ mm. Note that there are 500 (obtained from 100/0.2) layers to print the CUBE model because of the layer thickness $\sigma_0 = 0.2$ mm. Fig. 18(a)-(d) presents four different tool paths. These use only G1-codes and were generated by a program installed on the commercial printer used for the experiments in this work (The 3DWOX Desktop program [40]).

Contrarily, the tool path generated by our proposed method `HollowTron`, which utilizes G2/G3-codes as much as possible, is shown in Fig. 18(e). The 1st row in this figure includes the direction-parallel offset tool path, DirPrl, produced by 3DWOX Desktop (Fig. 18(a)). The 1st column in the figure provides a graphical visualization of the tool path, while a close-up is given in the 2nd column. The 3rd column provides a histogram showing the frequency distribution of G-codes, with respect to the length of each code shown on the horizontal axis. The colors in the histogram correspond to G-codes with specified length intervals. Red denotes codes shorter than 2 mm, such as <2 mm, blue indicates <4 mm, black indicates <20 mm, and green indicates >20 mm. For clarity, the red, blue, and green line segments in the visualized tool paths correspond to the red, blue, and green colors, respectively, in the histogram. The filled and unfilled bars in the

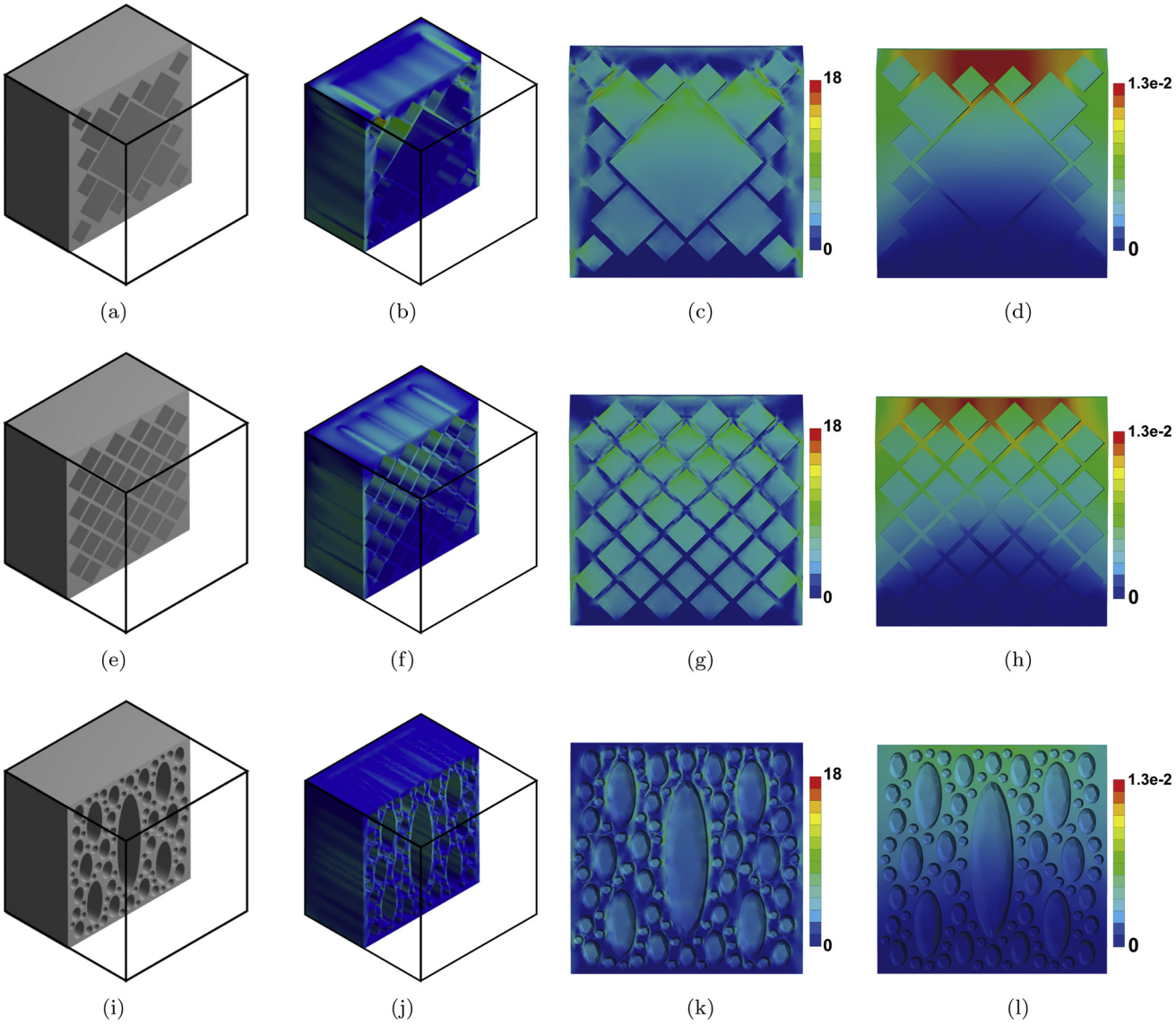


Fig. 12. The analysis of stress and deformation of cubes hollowed by three different methods. In all cases, pressure was uniformly applied to the cube tops. All analysis was done using ANSYS. 1st column: CAD model. 2nd and 3rd column: von Mises stress distributions. 4th column: Total deformation results. The stress and deformation results were normalized by the applied pressure (5 MPa) and the height of the cube (100 mm), respectively. (a) Sliced view of the cube hollowed by various sized rhombic holes, with a material infill density $\rho = 0.52$). (b) and (c) The isometric and sectional views, respectively, of the von Mises stress distribution corresponding to the structure in (a). (d) Total deformation for the structure shown in (a). (e–h) Hollowed by congruent rhombic holes ($\rho = 0.52$). (i–l) Hollowed by elliptic cylinders created by [14] ($\rho = 0.53$).

histogram represent G1-codes that extrude material and G0-codes that do not extrude material, respectively. These are shown as traversing line segments in Fig. 18(a-2). Clearly, the direction-parallel tool path has many short G0- and G1-codes. The two black, non-decreasing curves represent the cumulative distribution of G-codes. The lower solid curve corresponds to the filled bars (G1-codes) and the upper broken curve corresponds to the summation of both filled and unfilled (G1- and G0-codes). This comparison clearly illustrates the large number of G0-codes.

The 2nd row in the figure shows the contour-parallel offset tool paths, CntPrl-L, produced by 3DWOX Desktop (Fig. 18(b)). Here, circular features are first approximated by a number of line segments before tool path generation is attempted. Notably, most G-codes are short in length. The red ellipse in Fig. 18(b-2) corresponds to an unfilled region in the printed layer, and clearly this method leaves several of these small unfilled regions. The 3rd and 4th rows in the figure correspond to two other tool paths generated by 3DWOX Desktop: crystaline infill (Cryst) and grid infill (Grid), respectively. These methods leave unfilled regions. The 5th row (Fig. 18(e)) shows the contour-parallel offset tool path obtained using the Voronoi diagram (CntPrl-VD) produced by HollowTron, which implemented the proposed algorithm to take advantage of spheroid hollows. In general, all the generated tool paths were cross-validated using NCViewer [41].

Fig. 19 provides a summary of the histograms in the rightmost column of Fig. 18. The unfilled, gray, and hatched bars correspond to G0-, G1-, and G2/G3-codes, respectively, and the black filled bar denotes the total number of G-codes. Notably, our proposed method uses the smallest number of G-codes, which corresponds to a relatively fast, easy, and cost-effective process.

Fig. 20 shows the printed results of each layer generated by the tool paths shown in Fig. 18. The DirPrl, CntPrl-L, Crystl, and Grid methods are shown in Fig. 20(a)–(d), respectively. Interestingly, Fig. 20(b) contains non-printed void defects, which result from incomplete tool paths, as shown in Fig. 18(b-2). Previously, Kulkarni and Dutta [42] distinguished two distinct types of voids: intra-layer voids, which exist within a layer, and inter-layer voids, which exist between two adjacent

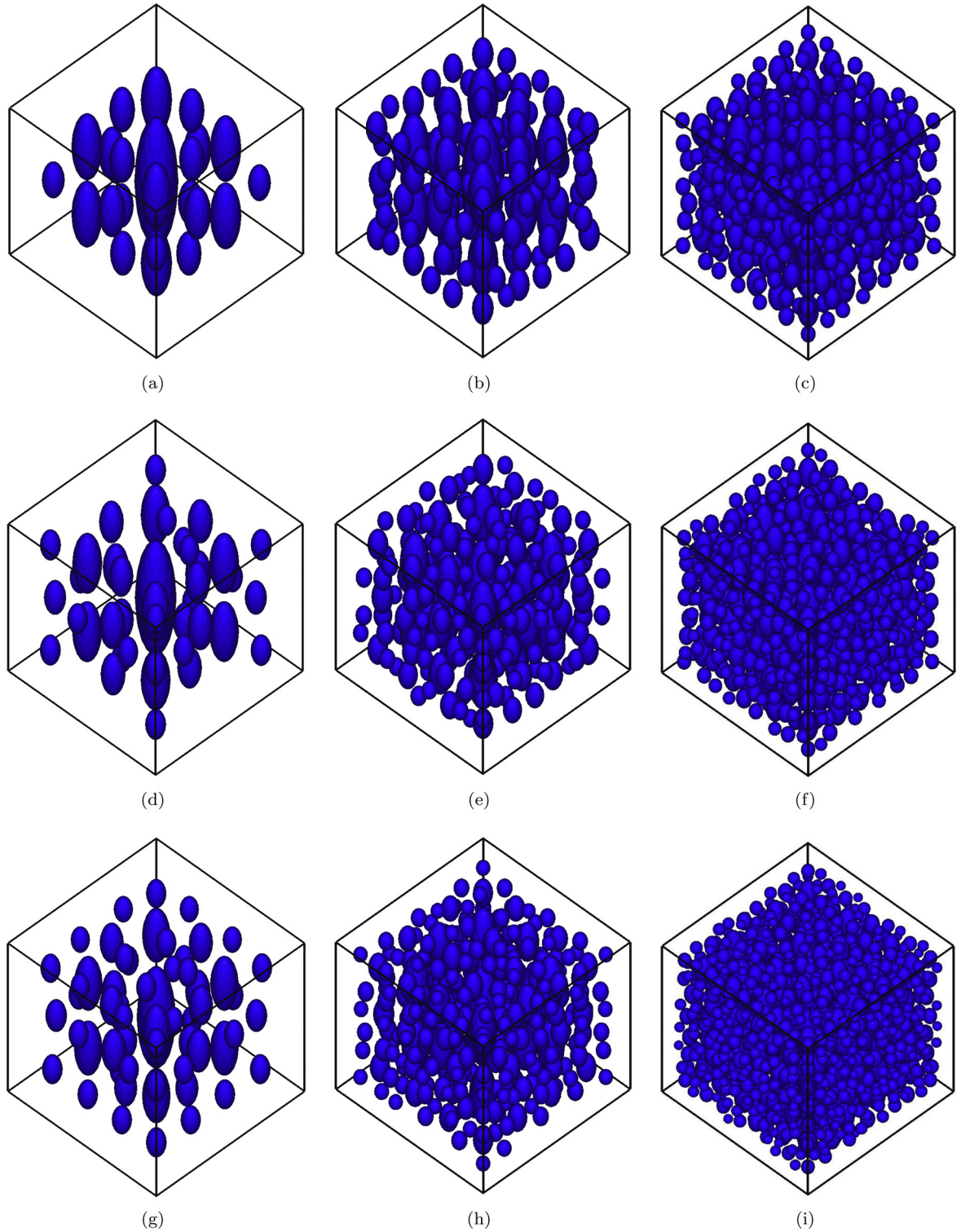


Fig. 13. Cube models with spheroidal hollows, generated by HollowTron, with different combinations of shrinkage ξ and infill density ρ. 1st column: $\rho = 0.9$. 2nd column: $\rho = 0.8$. 3rd column: $\rho = 0.7$. 1st row: $\xi = 0.9$. 2nd row: $\xi = 0.8$. 3rd row: $\xi = 0.7$. (a) CUBE($\xi = 0.9$, $\rho = 0.9$), (b) CUBE(0.9, 0.8), (c) CUBE(0.9, 0.7), (d) CUBE(0.8, 0.9), (e) CUBE(0.8, 0.8), (f) CUBE(0.8, 0.7), (g) CUBE(0.7, 0.9), (h) CUBE(0.7, 0.8), and (i) CUBE(0.7, 0.7).

layers and are critical for mechanical properties [42,43]. The voids in Fig. 20(b) provide a typical example of intra-layer voids. Fig. 20(c) and (d) show some non-printed infill patterns.

Fig. 20(e) presents the printed result of the CntPrl-VD tool path produced by `HollowTron` in Fig. 18(e). This print was done on the same printer, 3DWOX 1X, by simulating each circular motion with a set of short linear motions, obtained by approximating each corresponding arc with a set of line segments. Specifically, the print-out is produced by approximating an arc β with a line segment, where β has an angle 1° at its center, and one last segment has a smaller angle. All extrusion rates are identical for the structures shown in (a–e). This simulation approach had to be used in this study because the printer nozzle used could only move linearly. In addition, we found neither a printer nor printing service that could utilize circular printing motions with a

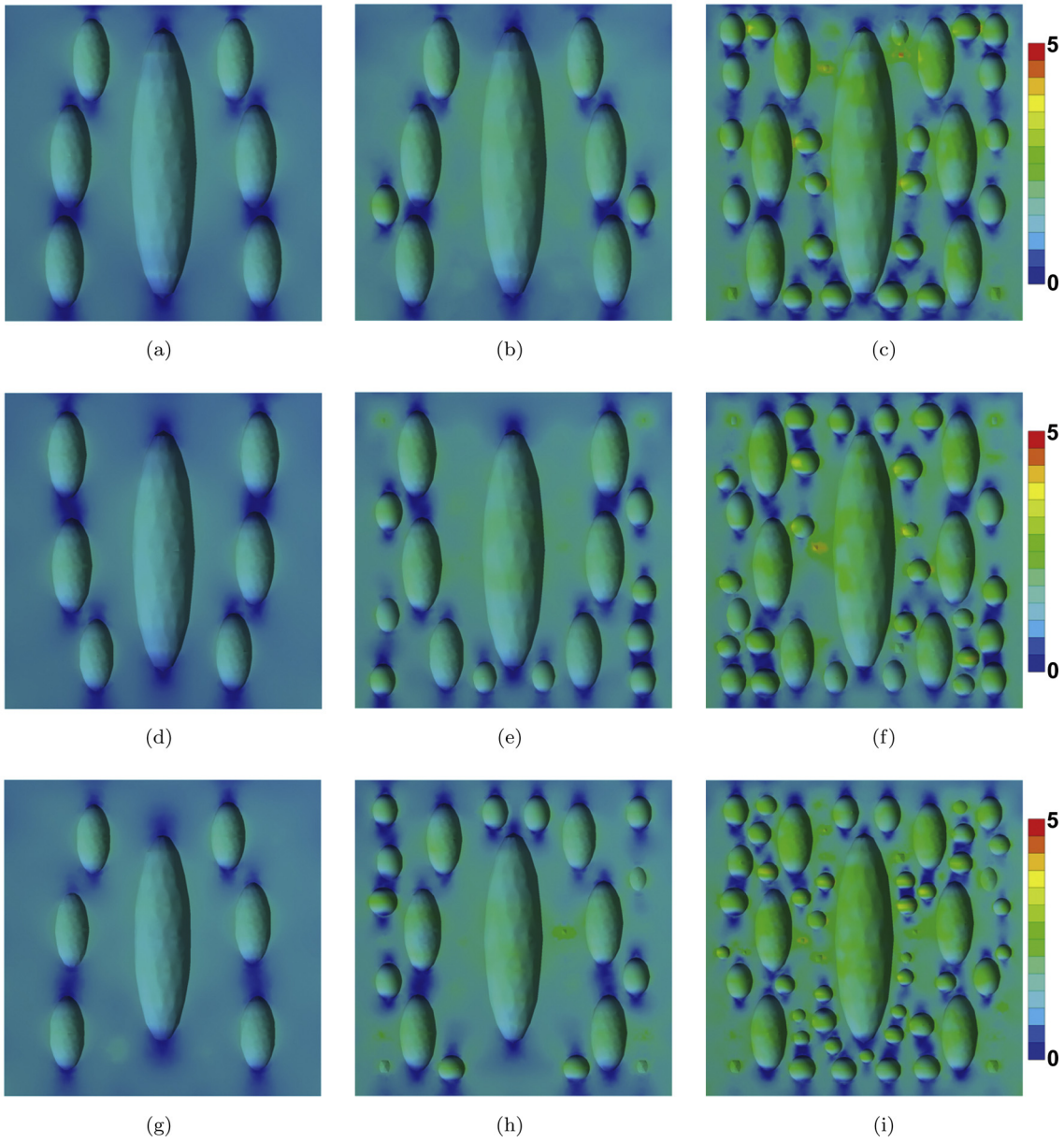


Fig. 14. Stress analysis results, visualized at the middle section plane. The images of the von Mises stress distribution are obtained from the section plane $X = 50$ mm. The results were normalized by the 5 MPa applied pressure. 1st column: $\rho = 0.9$. 2nd column: $\rho = 0.8$. 3rd column: $\rho = 0.7$. 1st row: $\xi = 0.9$. 2nd row: $\xi = 0.8$. 3rd row: $\xi = 0.7$. (a) CUBE($\xi = 0.9$, $\rho = 0.9$), (b) CUBE(0.9, 0.8), (c) CUBE(0.9, 0.7), (d) CUBE(0.8, 0.9), (e) CUBE(0.8, 0.8), (f) CUBE(0.8, 0.7), (g) CUBE(0.7, 0.9), (h) CUBE(0.7, 0.8), and (i) CUBE(0.7, 0.7).

Table 1
Summary of the numerical simulation results shown in Fig. 14 using CUBE models evaluated with ANSYS. Note, the stress and deformation values were normalized by the applied pressure (5 MPa) and the cube height (100 mm), respectively.

		ρ					
		0.9		0.8		0.7	
		Max. stress	Max. deform.	Max. stress	Max. deform.	Max. stress	Max. deform.
ξ	0.9	1.74	1.74e–2	2.38	2.05e–2	4.85	2.58e–2
	0.8	1.70	1.71e–2	2.62	2.07e–2	4.24	2.59e–2
	0.7	1.80	1.73e–2	3.15	2.12e–2	4.08	2.64e–2

reasonable cost. Notably, the printed results exhibit almost no void space. In fact, the tiny unfilled spots observed can be easily filled by tiny linear moves, which can be directly identified using the Voronoi edges.

Table 2 summarizes the analysis statistics for the five tool paths in Fig. 18. Column A analyzes the direction-parallel tool path (1st row of Fig. 18), which has a total of 2606 G-codes, including 1,417 G1-codes and 1,189 G0-codes, with a total length of 20,600 mm, composed of 19,186 and 1414 mm of G1- and G0-code, respectively. The total printing time was found to be 17 min and 14 s, with the following feed rates ϕ: $\phi_{G0} = 130$ mm/s for G0-codes and $\phi_{G123} = 20$ mm/s for G1-, G2-, or G3-codes. Assuming that the extruder moves at its feed rate, the time required to print all G1 and G0-codes was 15 min and 59 s (19,186/20) and 11 s (1,414/130), respectively. Notably, the sum of these times, 16 min and 10 s, is 1 min and 4 s shorter than the measured time of 17 min and 14 s. Thus, this represents a significant underestimation of the measured total printing time. This result is consistent with earlier

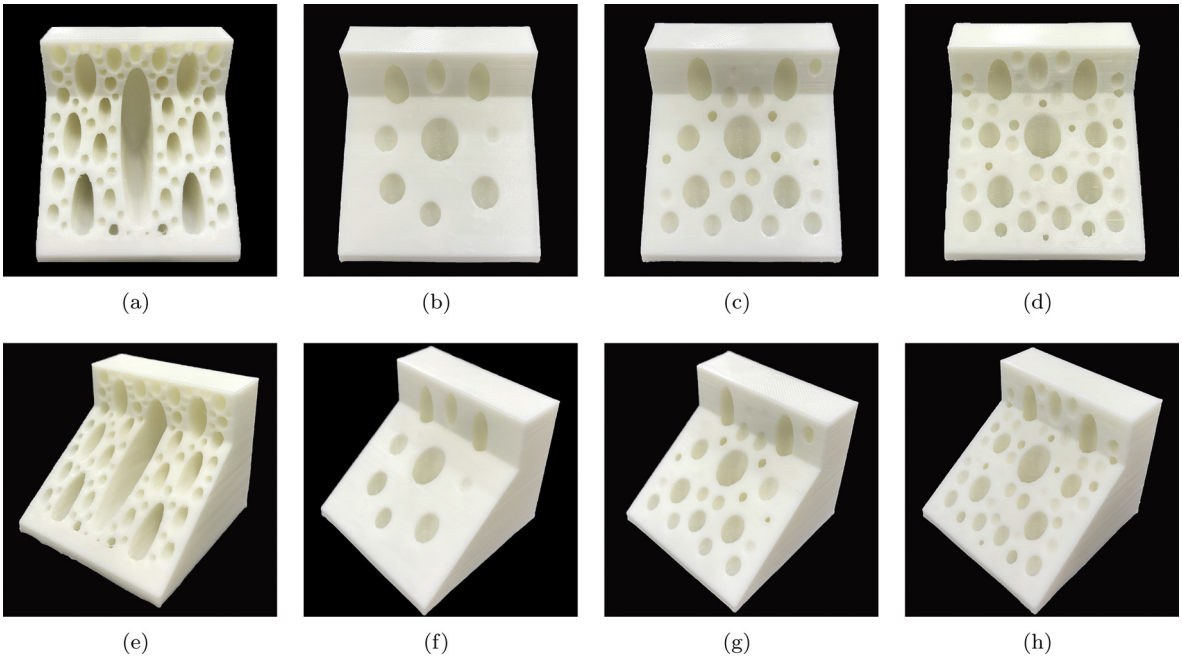


Fig. 15. Various printed hollowed cube models. Each cubes was cut by two planes to reveal their interior. Front and isomeric views are provided in the upper and lower rows, respectively. (a) and (e) Elliptic cylindrical hollows (Shrinkage $\xi = 0.9$, infill density $\rho = 0.5$). (b–d) and (f–h) Spheroidal hollows. (b) and (f) CUBE($\xi = 0.9$, $\rho = 0.9$). (c) and (g) CUBE(0.9, 0.8). (d) and (h) CUBE(0.9, 0.7).

works for NC milling machines [44], which also found that estimating machining time by the tool traveling distance and feed rate underestimated the real measured machining time. This discrepancy can be attributed to the acceleration and deceleration required for each head movement, which make contribute significantly to the machining time. Assuming that the time difference Σ of 1 min and 4 s is equally distributed throughout the 2,606 G0- and G1-codes, the acceleration and deceleration for each motion can be said to take 0.0246 s (γ6 and A1). The same analysis was performed for the contour-parallel CntPrl-L method, and the acceleration and deceleration in this case was found to take 0.0073 s on each step (γ6 and B1).

The same analyses for the other two methods revealed acceleration and deceleration times of 0.0269 and 0.0233 s, respectively, which are quite close to that of DirPrl (γ6). Note that the machining time is consistently underestimated in all the evaluated methods (γ2).

The average of Σ is 0.01595 s ((0.0246 + 0.0073)/2) for each pair of acceleration and deceleration steps. In this calculation, columns C1 and D1 were ignored, because these methods do not fill the entire layer but rather leave significant unfilled infills and thus have time characteristics similar to those in A1.) This Σ value was then applied to each of the 1,323 G-codes in the proposed method to generate a tool path, CntPrl-VD, with a total resulting time of 15 min and 15 s. This is an 11.5% and 12.6% reduction in printing time compared to the direction-parallel method, and the contour-parallel after linearization CntPrl-L, respectively. Notably, CntPrl-VD uses the fewest overall G-codes, and thus has the least acceleration-deceleration pairs with each extruder movement. Thus, this property provides a significant increase to the printing efficiency achieved by this method. It is determined that CntPrl-VD is superior to the direction-parallel path in AM, which contradicts the previous finding reported in [44] for a milling machine. It is expected that a spiral tool path, which can be derived from the CntPrl-VD tool path with some additional effort, will further increase its printing efficiency, as has been previously observed and reported in [42].

Reducing the total printing time can increase the productivity during AM printing processed, so it is desirable to: (i) shorten the total traveling distance of the extruder printing motions, (ii) reduce the turns taken between consecutive extruder printing motions, and (iii) reduce the total traveling distance of extruder non-printing motions, called bridging motions.

The advantage of using G2/G3-codes, as much as possible, is most impactful for design features such as hollow cylinders and hollow spheres, as shown in Fig. 21(a) and (b), respectively. For such features, the region to print on a typical layer is a hollow circle, which consists of two concentric circles, represented by the black solid circles in Fig. 21(c). The CntPrl-VD tool path is also shown, where red solid

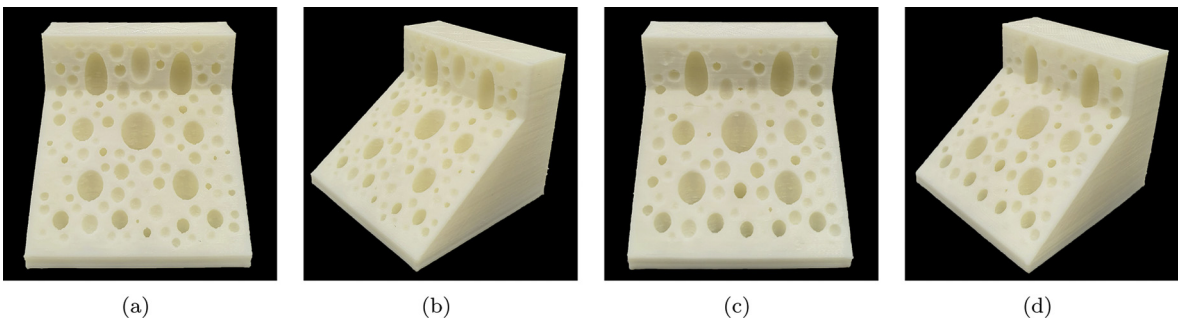
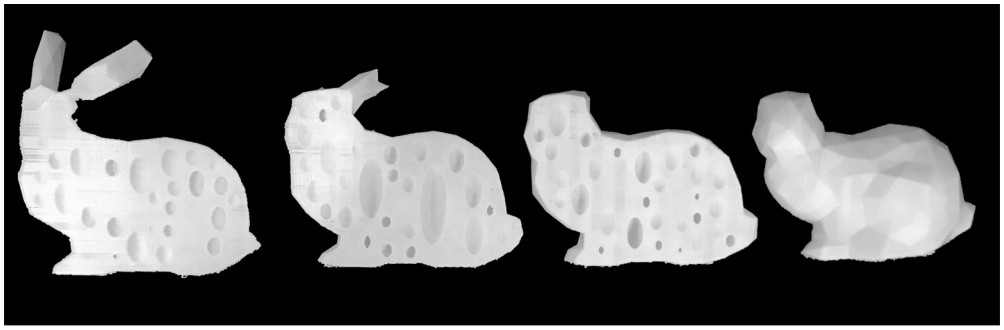
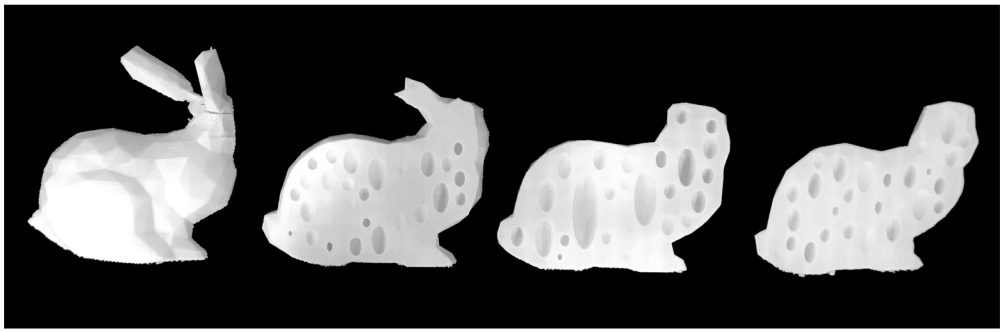


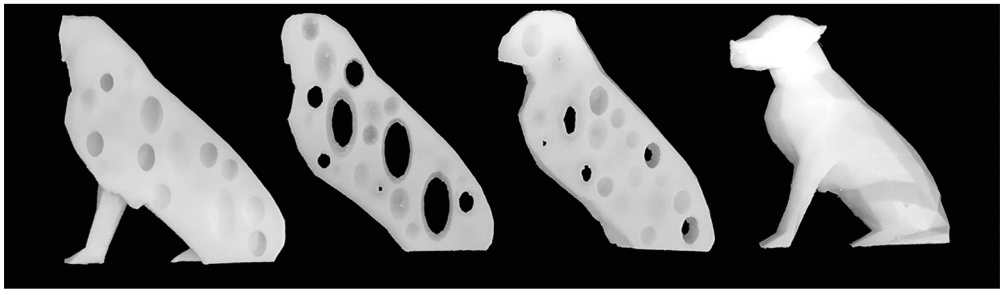
Fig. 16. Hollowed cube models with the same infill density ($\rho = 0.7$) but different shrinkage (ξ= 0.7 and 0.8). (a) and (b) CUBE($\xi = 0.7$, $\rho = 0.7$). (c) and (d) CUBE($\xi = 0.8$, $\rho = 0.7$).



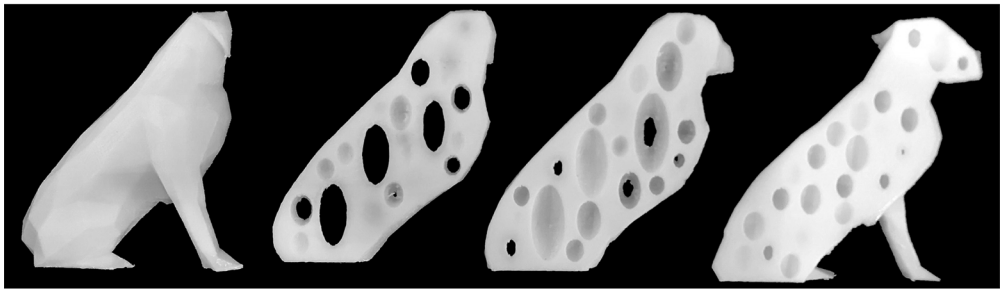
(a)



(b)



(c)



(d)

Fig. 17. Printed bunny and dog, with 500 and 300 faces, respectively, on the model boundary. The artifacts here were subdivided into four pieces by three section planes to reveal the interior structure. (a) and (b). BUNNY($\xi = 0.9$, $\rho = 0.8$). (c) and (d). DOG($\xi = 0.9$, $\rho = 0.8$).

circles and dotted arrows correspond to circular printing motions and non-printing motions, respectively. Fig. 21(d) shows the DirPrl tool path. Notably, the CntPrl-VD tool path has no sharp turn at all.

From the Voronoi diagram perspective, a hollow circle is a degenerate case, as shown in Fig. 21(e). In this case, the Voronoi diagram has a single V-edge, which contains no V-vertex where the V-edge is also a circle that is concentric with the generator circles. The case where the V-edge is an ellipse and the two generator circles are not concentric is shown in Fig. 21(f).

This experiment proves that the tool path utilizing G2/G3-codes, generated by the offsetting with the Voronoi diagram, has the following three advantageous properties: shorter printing time, better printed quality, and smaller file size.

4. Conclusions

In this paper, novel algorithms were proposed for generating support-free prolate spheroidal hollows, and corresponding tool paths utilizing G2/G3-codes were developed for AM applications. Contrary to currently popular methods based on rhombuses or cubes, spheroidal

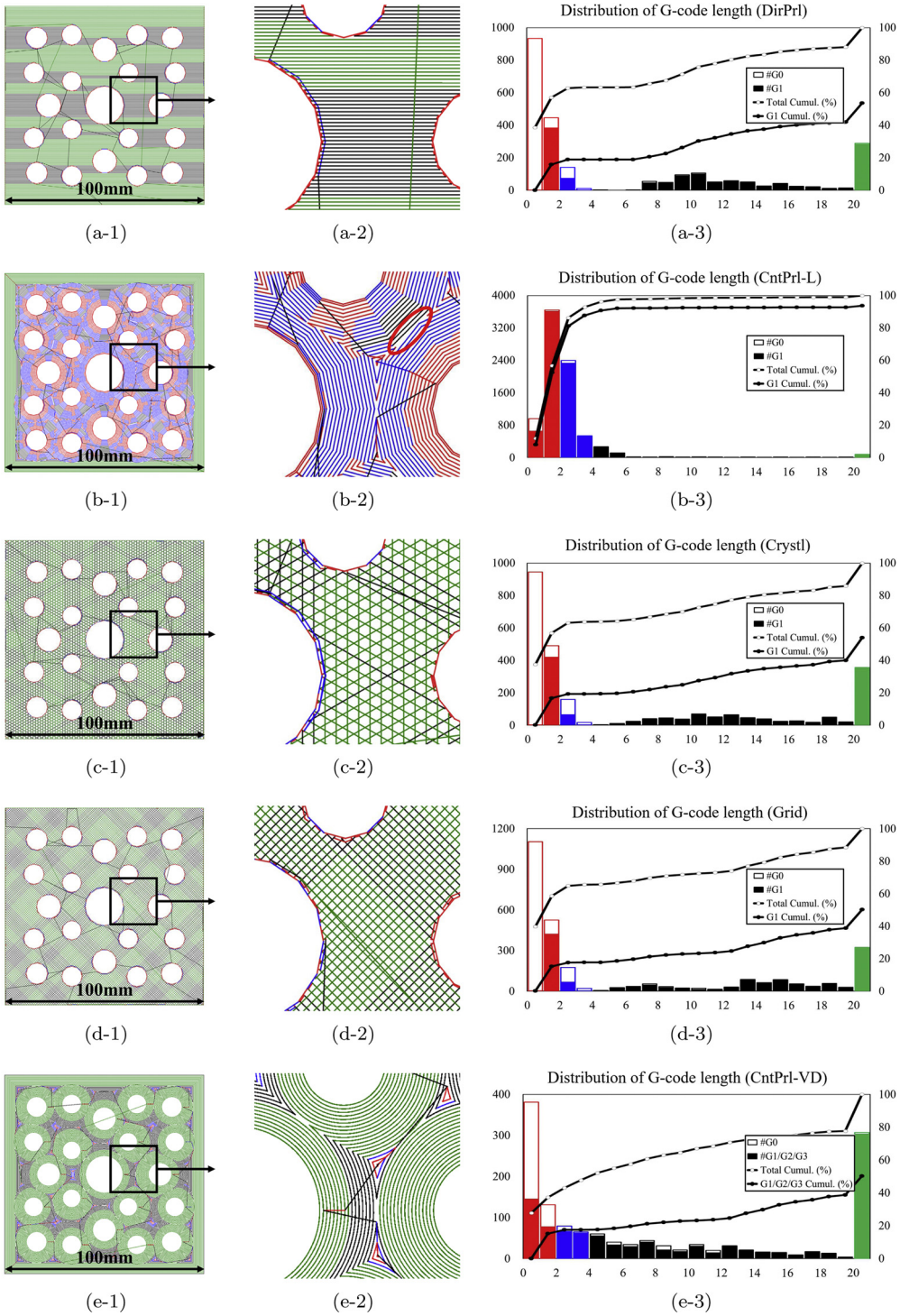


Fig. 18. The five different tool paths used and corresponding statistics. The data is the same as that used in Fig. 8(e), where the mid-plane $z = 50$ mm of CUBE($\xi = 0.7$, $\rho = 0.9$). The images in rows 1–4 were produced by 3DWOX Desktop, and the image in row 5 was produced by HollowTron. 1st row (a): direction-parallel (DirPrl), 2nd row (b): linearized contour-parallel (CntPrl-L), 3rd row (c): crystalline infill, 4th row (d): grid infill, and 5th row (e): contour-parallel offset (CntPrl-VD). Left column: tool path covering the entire plane, center column: close-up of the path shown in the left column, and right column: frequency distribution of tool motions with respect to code length. The colors were coded according to the length of code segments as follows. Red: < 2㎜, blue < 4㎜, black <20 mm, and green >20 mm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

hollows avoid sharp corners and the corresponding problem of high stress concentration locations. The resulting 3D shapes could be directly printed with extrusion-based and digital light processing-based 3D printers, without any extra interior supports. The tool paths fully utilize circular printing motions based on G2/G3-codes, and thus benefit from the associated advantages of shorter printing times, increased printing quality, and reduced tool path file sizes. As needed, two efficient algorithms, which were based on the topology-oriented incremental approach, were developed: one for constructing the Voronoi diagram of a polyhedron, and another for the Voronoi diagram of spheroids within a polyhedron, which to the best of our knowledge is the first of its kind. Experimental results confirm the proposed method performs well and is

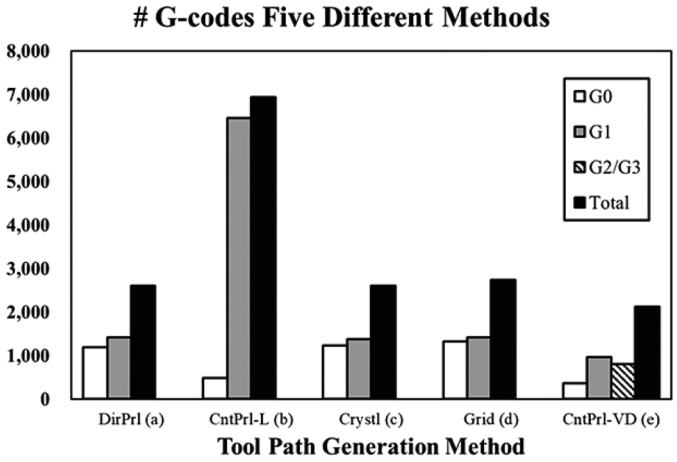


Fig. 19. A summary of Fig. 18, providing the number of G-codes for the five different tool paths, based on the same data used in Fig. 8(e). Interestingly, the proposed CntPrl-VD method uses the smallest number of G-codes. (a) DirPrl method (the 1st row in Fig. 18). (b) CntPrl-L method (the 2nd row in Fig. 18). (c) Crystl method (the 3rd row in Fig. 18). (d) Grid method (the 4th row in Fig. 18). (e) CntPrl-VD method (the 5th row in Fig. 18). White, grey, hatched, and black colors correspond to G0-codes, G1-codes, G2/G3-codes, and the total summation of codes, respectively.

practical for a variety of applications.

Limitation and future works. Our research provides a starting point for many future study directions. First, the hollows arrangement can be further optimized by controlling certain parameters, including the size and location of spheroids, the eccentricity of spheroids, and others. In this way, the material distribution in objects can be better controlled. Second, the method developed for spheroids could likely be generalized to other shapes, such that a given design responds to external forces, in addition to gravity, while satisfying all relevant printing constraints. Third, HollowTron can be improved to better accommodate user functionalities.

Authors' contribution

Seunghwan Choi: software, validation, formal analysis, writing – review and editing, visualization. Joonghyun Ryu: software, formal analysis. Mokwon Lee: software. Jehyun Cha: software. Hyunwoo Kim: validation, visualization. Chanyoung Song: validation, visualization. Deok-Soo Kim: conceptualization, methodology, writing-original draft, writing-reviewing and editing, project administration, funding acquisition.

Conflict of interest

None declared.

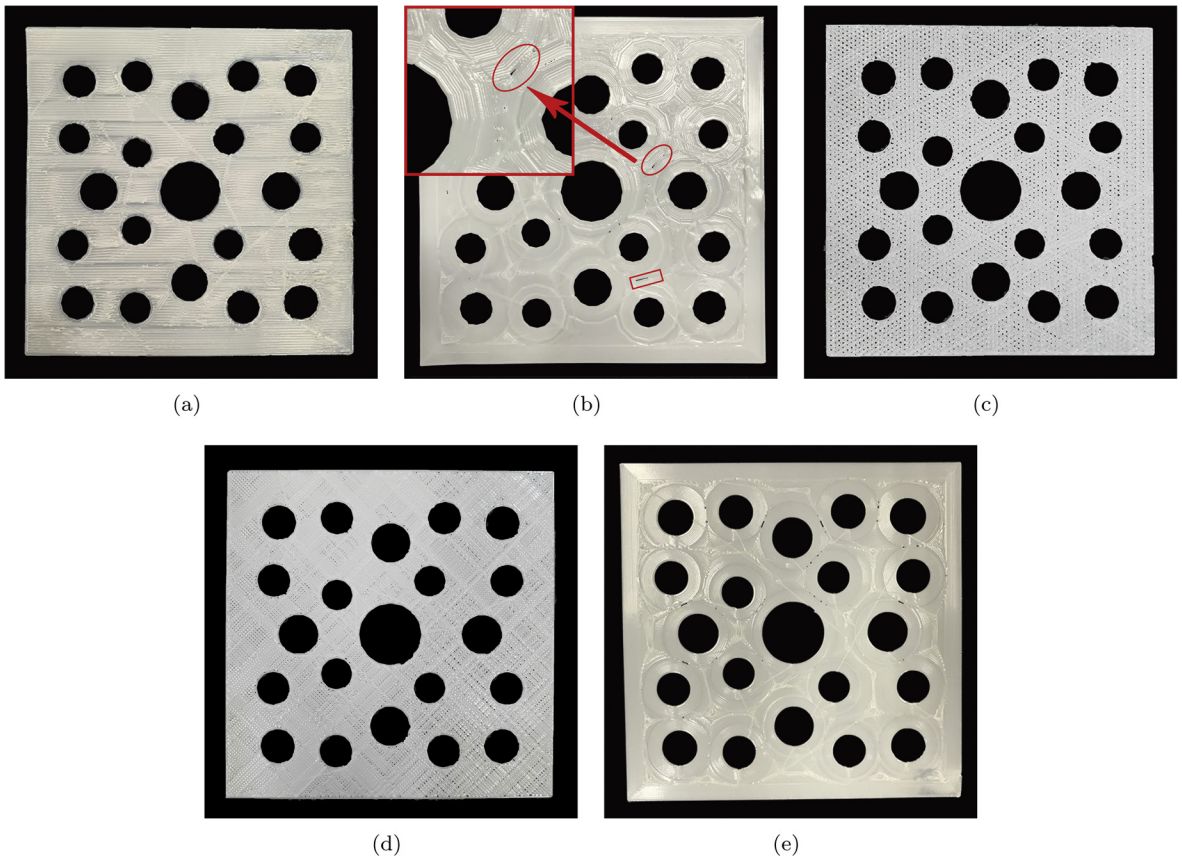


Fig. 20. Printed single-layer artifacts in Fig. 18 produced via different tool paths. The tool paths for (a) through (d) were generated using conventional tool path generation methods and printed by 3DWOX 1X. The bed temperature was set as 65 °C because no raft was used to print these layers. (a) Direction-parallel (DirPrl, 1st row in Fig. 18). (b) Contour-parallel, after linearization (CntPrl-L, 2nd row in Fig. 18). The corresponding unfilled regions in both figures are important to note in this case. (c) Crystalline (Crystl, 3rd row in Fig. 18). (d) Grid (4th row in Fig. 18). Remaining unfilled infills can be clearly observed in (c) and (d). (e) CntPrl-VD (5th row in Fig. 18). Each circular motion is simulated by a set of linear motions, so each corresponding arc β is approximated by a line segment, where β has an angle 1° at the center. The extrusion rate was kept consistent for all structures in (a–e).

Table 2
Comparison between theh five different tool paths: Direction-parallel (DirPrl), Linearized Contour-parallel (CntPrl-L), Crystalline (Crystl), Grid (Grid), and Contour-parallel offset using Voronoi diagram (CntPrl-VD). DirPrl, CntPrl-L, Crystl, and Grid were generated by 3DWOX Desktop, and CntPrl-VD was generated by HollowTron. The tool paths are analyzed for three criteria: number of G-codes, length of G-codes, and total printing time. The G-code length unit is mm.

Offset method				DirPrl(A) (A1)		CntPrl-L (B) (B1)		Crystl (C) (C1)		Grid (D) (D1)		CntPrl-VD (E) (E1)	
				#	(%)	#	(%)	#	(%)	#	(%)	#	(%)
# G-code	Grand total		(α1)[a]	2,606	(100)	6,941	(100)	2,602	(100)	2,743	(100)	1,323	(100)
	Printing	Total	(α2)[b]	1,417	(54)	6,459	(93)	1,371	(53)	1,416	(52)	964	(73)
		G1	(α3)	1,417	(54)	6,459	(93)	1,371	(53)	1,416	(52)	164	(12)
		G2 & G3	(α4)	-	(0)	-	(0)	-	(0)	-	(0)	800	(61)
	Bridging	G0	(α5)	1,189	(46)	482	(7)	1,231	(47)	1,327	(48)	359	(27)

				(A2)		(B2)		(C2)		(D2)		(E2)	
				Length	(%)	Length	(%)	Length	(%)	Length	(%)	Length	(%)
Length G-code	Grand Total		(β1)[c]	20,600	(100)	20,956	(100)	20,304	(100)	20,643	(100)	18,659	(100)
	Printing	Total	(β2)[d]	19,186	(93)	19,691	(94)	18,787	(93)	19,181	(93)	17,888	(96)
		G1	(β3)	19,186	(93)	19,691	(94)	18,787	(93)	19,181	(93)	4,569	(25)
		G2 & G3	(β4)	-	(0)	-	(0)	-	(0)	-	(0)	13,319	(71)
	Bridging	G0	(β5)	1,414	(7)	1,265	(6)	1,517	(7)	1,462	(7)	771	(4)

				(A3)	(B3)	(C3)	(D3)	(E3)	
Printing time	Measured		(γ1)	17 min 14 s	17 min 25 s	17 min 0 s	17 min 14 s	15 min 15 s	
								Saved(A) 11.5%	Saved(B) 12.6%
	Estimated[f]	Total	(γ2)[e]	16 min 10 s	16 min 34 s	15 min 50 s	16 min 10 s	15 min 0 s	
		G1	(γ3)	15 min 59 s	16 min 24 s	15 min 39 s	15 min 59 s	3 min 48 s	
		G2 & G3	(γ4)	-	-	-	-	11 min 6 s	
		G0	(γ5)	11 s	10 s	11 s	11 s	6 s	
	(De)accel. time/G-code		(γ6)	0.0246 s	0.0073 s	0.0269 s	0.0233 s	0.0160 s	

[a] (α1) = (α2) + (α5).
[b] (α2) = (α3) + (α4).
[c] (β1) = (β2) + (β5).
[d] (β2) = (β3) + (β4).
[e] (γ2) = (γ3) + (γ4) + (γ5).
[f] The feed rates (ϕ) for G1, G2, and G3: 20 mm/s, G0: 130 mm/s.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF), with a grant funded by the Korea government (MSIP, MSIT) (Nos. 2017R1A3B1023591,2016K1A4A3914691). The authors additionally thank the anonymous reviewers for their constructive comments and suggestions. We would like to thank Professor Gilho Yoon for technical advice and Editage (www.editage.com) for English language editing.

Appendix A. Literature review on modeling for 3D printing

Interior hollowing. The initial proposal for detect structural defects in fabricated objects was made by Telea and Jalba (2011), as a notion of printability involving geometrical attributes like local thickness as well as the shape and topology of thin regions and their surroundings [45]. Stava et al. (2012) then introduced a method for detecting structural problems in fabricated objects causing them to break or collapse due to insufficient structural strength. In this work, they used a lightweight structural analysis solver, together with an approximation of 3D medial axis, and introduced a method to correct any regions with high structural stress by combining thickening and strut insertion operations to enhance hollow mechanical strength [4]. Wang et al. (2013) presented skin-frame structure method, which is used to minimize the number of struts by solving a 0-sparsity optimization problem formulated as a multi-objective programming problem [8]. The fabricated objects here were physically stable and could be printed with significantly less material. Lu et al. (2014) introduced the concept of support-free hollowing, using both simulation and physical evaluations of object strength, by arranging honeycomb-cells with an adaptive centroidal Voronoi diagram to define the inner structure [9]. Zhang et al. (2015) reported a method for designing internal frame structures based on a medial axis, such that objects were fabricated with a minimal amount of material but could still withstand a given external load [46]. Both static and dynamic balances were studied by designing hollows, interior infills, and dynamic model shapes, among other methods [47–50]. Lee and Lee (2017) proposed an algorithm, based on three-dimensional block partitioning, where the blocks were obtained by dividing objects via arbitrary planes [16]. Hornus et al. (2017) reported an iterated geometric carving algorithm, used to compute maximally large empty cavities with nested sets of inner walls, all of which had exactly the minimal printable thickness [49]. This algorithm was based on the medial axis of polygon interior and an exact polygon offsetting technique. Lee et al. (2018) presented an algorithm for creating hollows of elliptic cylinders, which were created by a projection of planar ellipses computed using the Voronoi diagram of planar circles [14]. Martinez et al. (2018) introduced a microstructure with a wide range of elastic behaviors, which can be fabricated by a fused filament fabrication process by enforcing continuity and self-support while satisfying the overhang angles constraint [50]. This structure offered a range of orthotropic elastic responses that could be spatially graded, providing a promising option for hollow infill patterns. Their method utilized the Voronoi diagram based on clever polyhedral distances, rather than the typical Euclidean distance. Zhou et al. (2019) presented an approach for

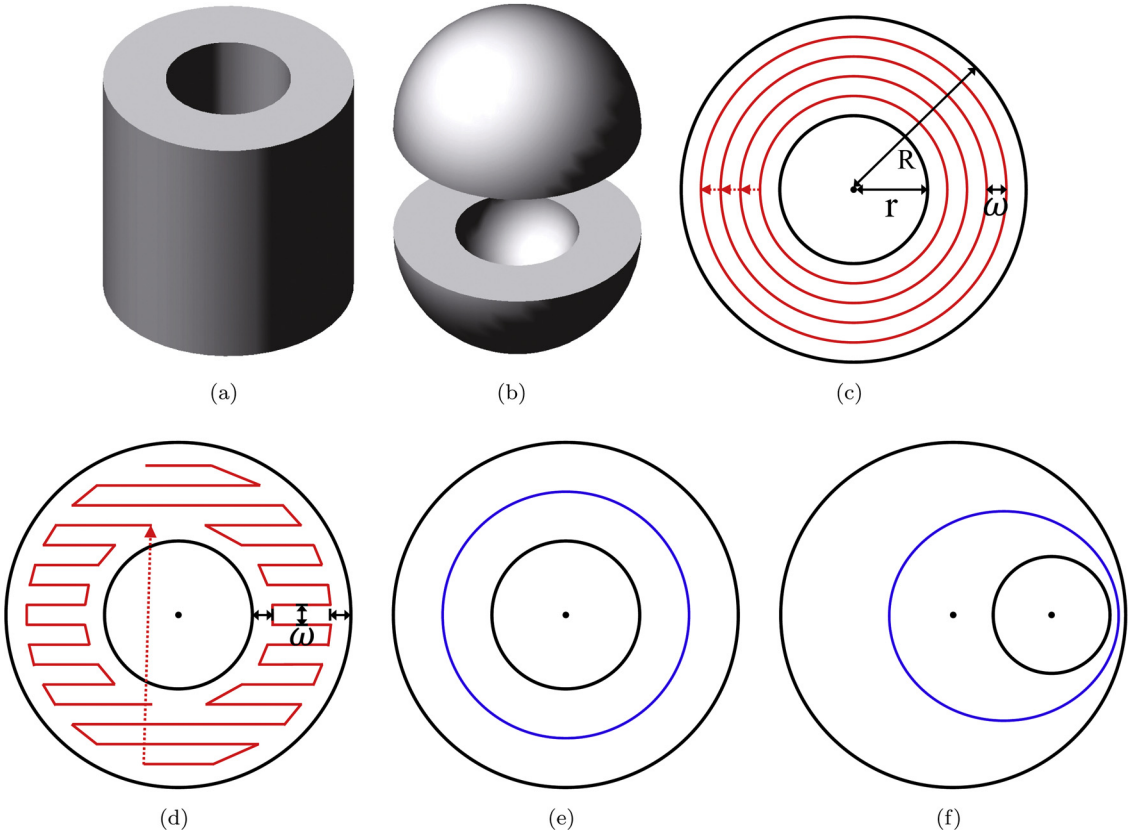


Fig. 21. Printing layer with the hollow circle shape, which is most effectively and efficiently produced by circular printing motions. (a) Hollow cylinder. (b) Hollow sphere. (c) Hollow circle with two concentric (solid black) circles with the CntPrl-VD tool path. (d) DirPrl tool path. The red solid circles and lines are printing motions, and the red dotted lines are non-printing motions. (e) Voronoi diagram of a hollow circle defined by two concentric circles. The blue solid curve is the V-edge, which is a concentric circle. (f) The Voronoi diagram of a hollow circle defined by two non-concentric circles. The blue solid curve is the V-edge, which is an ellipse. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

generating a tree-like support structure, similar to a pillar structure, which could be applied to both the exterior and interior of models [51]. Furthermore, Zhang et al. (2019) used barycenter in a similar way [52]. Tricard (2019) reported a greedy algorithm to place rib-like structures on hollow walls, with the intention of supporting the hollow roof [53]. The resulting structure formed a ribbed-vault, following a hierarchy of rib-like walls that were progressively eroded away and straightened, eventually merging with the interior object walls. Vaissier et al. (2019) proposed an algorithm that could create the geometry and topology of both inner and outer support structures by removing some initial lattice structure beams within the object exterior with a genetic algorithm [54]. Ulu et al. (2019) introduced a shape parameterization method used to design a minimum weight shell object, without a self-intersecting boundary [55]. This method can accommodate not only gravity, but also external forces from other directions, but may require additional interior support structures to be successfully fabricated. Thorough reviews on the process view of AM can be found at [56,57]. However, all methods suffer from the presence of sharp corners, with C^0-continuities on object boundaries.

Support-free structure. Support-free structures became a subject of study in 2016. Reiner and Lefebvre [12] proposed three interactive sculpting operators of trim, preserve, and grow, to be used for designing support-free models that require no support structures whatsoever. Wu et al. (2016) developed a method, based on topology optimization, to generate support-free infill structures, which produced a solid model minimizing compliance in the material update procedure in each iteration of the optimization process [15]. While this work considered the minimum thickness of the generated structure during the optimization process as a manufacturing constraint, it did not include the overhang constraint. Concurrent works of Langelaar [10] and Gaynor and Guest [58] (both in 2016), and the subsequent work by Johnson and Gaynor [59], also relied on topology optimization, but these considered an overhang angle threshold during the optimization process to generate support-free material distributions. Unfortunately, none of these approaches were directly connected to those utilizing interior hollows. Wang et al. (2019) presented an optimization method to reduce the total material consumed by printed objects and their support structures by minimizing the outer support volume of a printed model [60]. In this work, they built a self-supporting frame with a set of scale-adaptive parallelepiped grids to replace the solid interior of the printed model, so that the material consumption could be further reduced. Jiang et al. (2019) studied how printing orientation could influence the productivity, in relation to both interior and exterior support structures [61]. The objective here was to reduce the total amount of material consumed, as well as the production time and energy required for object manufacturing. Many prior works that utilize topology optimization techniques, applied to support structure design, did not address the relevant problem of avoiding large overhangs [62]. Furthermore, all previously reported methods leave C^0-continuities on object boundaries.

Appendix B. Brief review on tool path generation for 3D printing

A 3D printer is an NC machine, and a tool path must be generated to drive an extruder to fill material on the model boundary and interior for each layer. Generally, this tool path is translated to a set of G-codes, which has been the standard for NC machines for machining since the 1950s [63].

Among others, G0, G1, G2, and G3 are the most critical commands, and they correspond to rapid positioning, linear, clockwise circular, and counterclockwise circular printing motions, respectively. Tool path generation is a critical step in the planning process for AM. This step depends directly on the tool path strategy, also referred to as deposition strategy or layer filling strategy [42].

There are four primary strategies for generating tool paths in AM: (i) direction-parallel method (also referred to as zigzag or raster method), (ii) contour-parallel method, (iii) spiral method, and (iv) Hilbert space-filling curve method. A direction-parallel zigzag tool path is easy to generate, and has thus been the most popular choice for NC milling machines [44,63]. In fact, most studies on tool path generation in AM also use this approach [64–66]. Further, possible warpage problems and the stacking of fault lines can be prevented by changing the direction of tool path from layer to layer [42,67]. However, the resulting tool path may contain many short tool motions, which can result in the fabricated artifact containing many tiny voids.

Contour-parallel tool paths have many advantages in AM applications. However, developing a robust algorithm to generate these tool paths is difficult, because of the challenge involved in constructing the Voronoi polygon diagram, which is the main computational building block for these tool paths. Hence, some studies have used a hybrid approach, using the offsetting-based contour-parallel tool path on the model boundary and the direction-parallel zigzag tool path for its interior [68]. Still, few reliable algorithms for constructing this type of Voronoi diagram are available [69,70,36,14]. One potential problem, from the perspective of AM, is the fault line stack-up that can occur when two consecutive layers are printed by identical tool paths. In this case, it may be beneficial to print the second layer by a tool path generated from a different method, to avoid this potential danger. Contour-parallel methods for AM were discussed in detail in [71–73]. Interestingly, [74] presented an idea for using the medial axis transformation of a polygon P, which is a subset of the Voronoi diagram of P, to generate the extruder tool path. They demonstrated this concept using a case study of a thin wall.

Contrarily, spiral paths are derived from a contour-parallel path, and are formed by trimming an unnecessary segment of one offset loop and connecting it to the adjacent offset loop. A spiral path avoids using unnecessary G0-codes as much as possible, and instead uses consecutive printing motions of G1-, G2-, and G3-codes. Hence, spiral paths are the most efficient option among all tool path generation methods for reducing printing time. Another popular tool path is the use of a Hilbert space-filling curve, a special type of fractal curve [75–79]. Unfortunately, this method produces unnecessarily many stops and turns in the tool motion.

We emphasize here that the contour-parallel and spiral tool paths are particularly important for printing a single-layer-of-parts nesting problem to build a mixed batch of homogeneous or heterogeneous parts. This is because the polygonal regions from a single part are printed by a single, or very few, connected printing motion(s), and long bridge motions are minimized [80–82]. To solve this problem, we must determine the build orientation for each part, solve the bin-packing problem of parts in the printing space, and generate the tool path. Notably, there are several studies on tool path generation from arcs, circles, Bezier curves, B-spline curves, and NURBS surfaces [83–88], among other approaches.

Appendix C. Brief review of related Voronoi diagrams

In this appendix, the types of Voronoi diagrams that are used extensively by the proposed method are briefly reviewed.

The Voronoi diagram (VD) of a set of generators is defined as the tessellation of space such that each tessellation cell contains the locations that are closer to the corresponding generator than to any others. In general, VD is known to be the most efficient and compact data structure for spatial reasoning among particles. Various VD types can be defined by generator type, distance definition, and dimensions, with further details provided in [89]. In this study, we used the VD of spherical balls and ellipsoids, particularly those constructed within a polyhedron $\mathcal{P}$ in 3D. Unless otherwise specified, all distances used in this paper are the Euclidean distances.

The ordinary Voronoi diagram of points. The most basic and popular VD is the ordinary Voronoi diagram VD(P) of a set P of generator points in 2D and 3D. In VD(P), V-edges are line segments and V-cells are convex. Fig. 22(a) provides an example VD(P) with 11 points in the plane.

The Voronoi diagram of a polygon in 2D. Let Poly2 be a simple polygon, defined by a set V of vertices and a set E of edges, which are denoted as p-vertices and p-edges, respectively. The Voronoi diagram of this polygon VD(Poly2) is defined according to the general definition of Voronoi diagrams. Fig. 22(b) shows an example VD(Poly2) in 2D where Poly2 is a 2D bunny polygon, produced from the 3D bunny model in Fig. 22(d). Each p-edge is associated with, and is the owner of, a V-cell, and each reflex p-vertex, defined as one with an internal angle between two incident p-edges greater than 180°, is also the owner of a V-cell. The efficient and robust construction of VD(Poly2) is not trivial, and there are only very few robust codes available for this [36,89,14]. The red circle in Fig. 22(b) represents the maximum clearance probe that can be found in $O(n)$ time from VD (Poly2).

The Voronoi diagrams of disks and balls. The Voronoi diagram, VD(D2), of a set D2 of circular disks is a generalization of the ordinary Voronoi diagram of points in 2D. Here, the distances are defined from the disk boundaries. Notably, VD(D2) has intriguing topological properties that do not exist in VD(P), the details of which can be found in [90,91]. In VD(D2), the V-edges are hyperbolic arcs, and each V-edge is curved so that the half-space of the smaller generator is convex. Therefore, is disks are different sizes, some V-cells are non-convex. Importantly, VD(D2) can readily be constructed correctly and efficiently [90–92,29]. An example of the Voronoi diagram VD(D2,Poly2) of a disk set D2, contained in a non-convex polygon Poly2 with 758 edges, which is obtained from the 3D Stanford bunny model, is presented in Fig. 22(c). Here, the red circle represents the maximum clearance probe after the five disks are inserted into the Voronoi diagram of the polygon.

Let VD(B3) be the VD of a set of 3D spherical balls, B3, and VD(B3,Poly3) be the VD of balls in the polyhedron Poly3 in 3D. Fig. 22(d) shows VD (B3,Poly3), where B3 has five spherical balls and Poly3 is the 3D bunny model, decimated to have 500 triangular faces due to the computational time constraint for experiments.

Voronoi diagram of ellipses and ellipsoids. For a given plane, the VD can be further generalized. Let VD(E2) be the Voronoi diagram of a set of ellipses, E2, where the distance is defined by the ellipse boundaries. In VD(E2), the V-vertex v defined by three ellipses corresponds to the roots of a polynomial with a degree as high as 184 [93], which makes it difficult to find the roots exactly and efficiently. The V-edge between two ellipses can also be very complicated to define [94]. Therefore, the computation of V-vertices, V-edges, and their association through the topological structure among ellipses in VD(E2) is a significant challenge. There are only very few studies on constructing such a VD(E2) [93,14]. Let VD(E2,Poly2) be the VD of E2 contained in a polygonal container Poly2 in 2D. To the best of our knowledge, no previous work besides our own has been done on such a VD [14]. Fig. 22(e) provides an example of VD(E2), where the ellipses are the bunny polygon. Let VD(E3) be the Voronoi diagram of a set E3 of ellipsoids in 3D, and VD(E3,Poly3) be the Voronoi diagram of E3 within the polyhedron Poly3 in 3D. Neither construction algorithms has been reported yet. An example of VD(E3,Poly3), where Poly3 is the bunny, and E3 has five ellipsoids, specifically spheroids, is shown in Fig. 22(f).

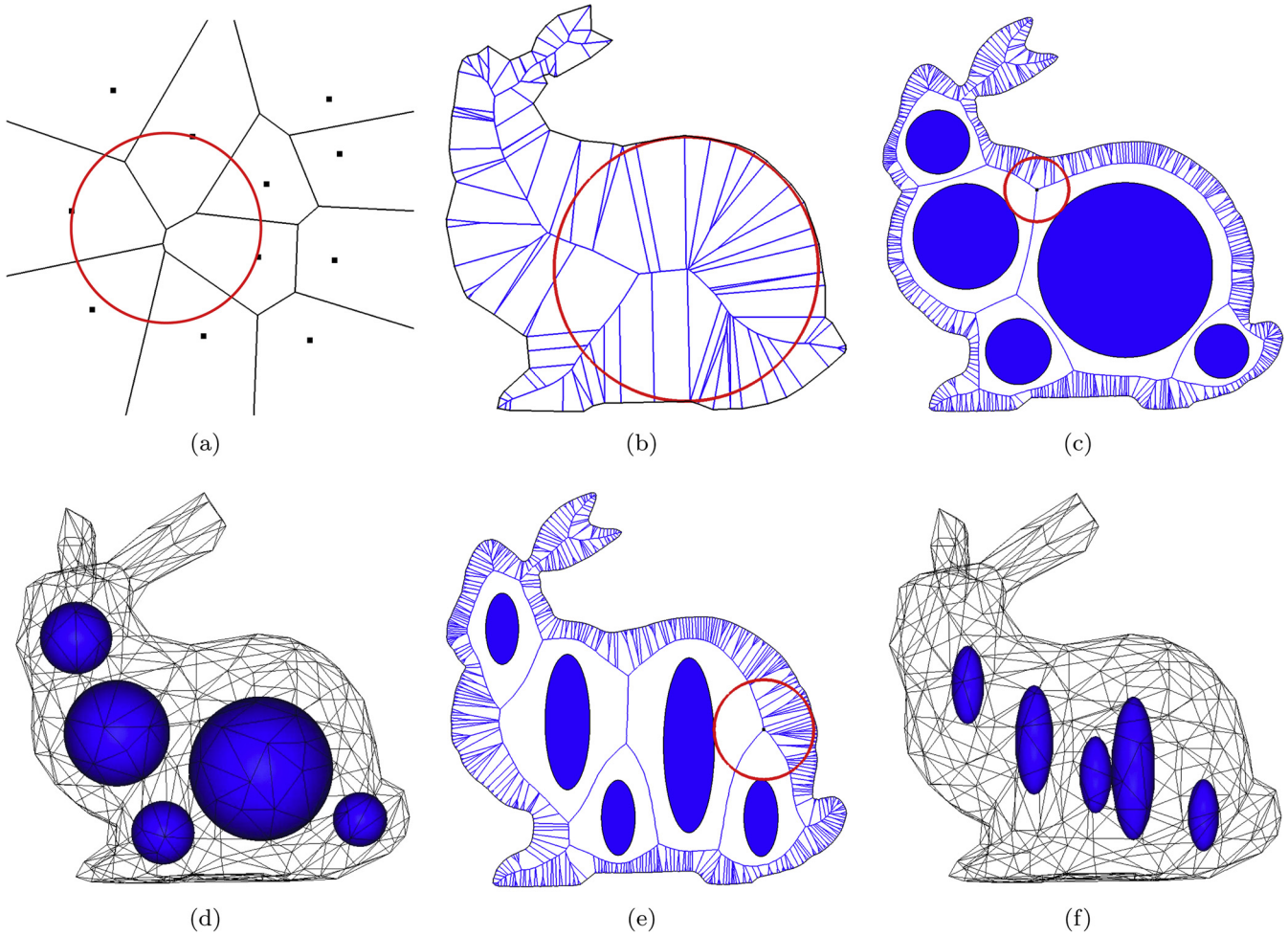


Fig. 22. Voronoi diagrams relevant to the algorithm proposed in this work. (a) Voronoi diagram VD(P2) of 11 points in 2D. The red circle represents the maximum clearance circle. (b) Voronoi diagram VD(Poly2) of a polygon Poly2 in 2D. (c) Voronoi diagram VD(D2,Poly2) of a set D2 of circular disks in Poly2, with 758 edges. (d) Voronoi diagram VD(B3,Poly3) of a set B3 of spherical balls in a polyhedron Poly3, with 500 triangular faces. (e) Voronoi diagram VD(E2,Poly2) of a set E2 of ellipses in Poly2. (f) Voronoi diagram VD(E3,Poly3) of a set E3 ellipsoids in Poly3. In this case, the ellipsoids are actually vertically-aligned prolate spheroids.

References

[1] G. Strano, L. Hao, R.M. Everson, K.E. Evans, A new approach to the design and optimisation of support structures in additive manufacturing, Int. J. Adv. Manuf. Technol. 66 (9–12) (2013) 1247–1254.

[2] J. Dumas, J. Hergel, S. Lefebvre, Bridging the gap: automated steady scaffoldings for 3d printing, ACM Trans. Graph. 33 (4) (2014) 98:1–98:10.

[3] J. Vanek, J.A.G. Galicia, B. Benes, Clever support: efficient support structure generation for digital fabrication, Comput. Graph. Forum 33 (5) (2014) 117–125.

[4] O. Stava, J. Vanek, B. Benes, N. Carr, R. Měch, Stress relief: improving structural strength of 3d printable objects, ACM Trans. Graph. 31 (4) (2012) 48:1–48:11.

[5] M.S. Jeon, Y. Jeon, J.H. Hwang, C.S. Heu, S. Jin, J. Shin, Y. Song, S.C. Kim, B.-K. Cho, J.-K. Lee, et al., Fabrication of three-dimensional porous carbon scaffolds with tunable pore sizes for effective cell confinement, Carbon 130 (2018) 814–821.

[6] M.S. Saleh, J. Li, J. Park, R. Panat, 3d printed hierarchically-porous microlattice electrode materials for exceptionally high specific capacity and areal capacity lithium ion batteries, Addit. Manuf. 23 (2018) 70–78.

[7] Z. Larimore, S. Jensen, P. Parsons, B. Good, K. Smith, M. Mirotznik, Use of space-filling curves for additive manufacturing of three dimensionally varying graded dielectric structures using fused deposition modeling, Addit. Manuf. 15 (2017) 48–56.

[8] W. Wang, T.Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3d objects with skin-frame structures, ACM Trans. Graph. 32 (6) (2013) 177:1–177:10.

[9] L. Lu, A. Shart, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, B. Chen, Build-to-last: Strength to weight 3d printed objects, ACM Trans. Graph. 33 (4) (2014) 97:1–97:10.

[10] M. Langelaar, Topology optimization of 3d self-supporting structures for additive manufacturing, Addit. Manuf. 12 (2016) 60–70.

[11] J. Wu, C.C.L. Wang, X. Zhang, R. Westermann, Self-supporting rhombic infill structures for additive manufacturing, Comput.-Aided Des. 80 (2016) 32–42.

[12] T. Reiner, S. Lefebvre, Interactive modeling of support-free shapes for fabrication, Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers (2016) 25–28.

[13] Y.-J. Liu, C. Xu, Y. He, D.-S. Kim, The duality of geodesic Voronoi/Delaunay diagrams for an intrinsic discrete laplace-beltrami operator on simplicial surfaces, Proceedings of the 26th Canadian conference on computational geometry (2014).

[14] M. Lee, Q. Fang, Y. Cho, J. Ryu, L. Liu, D.-S. Kim, Support-free hollowing for 3d printing via Voronoi diagram of ellipses, Comput.-Aided Des. 101 (2018) 23–36.

[15] J. Wu, C. Dick, R. Westermann, A system for high-resolution topology optimization, IEEE Trans. Visual. Comput. Graph. 22 (3) (2016) 1195–1208.

[16] J. Lee, K. Lee, Block-based inner support structure generation algorithm for 3d printing using fused deposition modeling, Int. J. Adv. Manuf. Technol. 89 (5–8) (2017) 2151–2163.

[17] W.D. Pilkey, D.F. Pilkey, Peterson's Stress Concentration Factors, John Wiley & Sons, 2008.

[18] D.K. Patel, A.H. Sakhaei, M. Layani, B. Zhang, Q. Ge, S. Magdassi, Highly stretchable and UV curable elastomers for digital light processing based 3d printing, Adv. Mater. (2017).

[19] Q. Mu, L. Wang, C.K. Dunn, X. Kuang, F. Duan, Z. Zhang, H.J. Qi, T. Wang, Digital light processing 3d printing of conductive complex structures, Addit. Manuf. 18 (2017) 74–83.

[20] L. Ge, L. Dong, D. Wang, Q. Ge, G. Gu, A digital light processing 3d printer for fast and high-precision fabrication of soft pneumatic actuators, Sens. Actuators A: Phys. 273 (2018) 285–292.

[21] A. Luongo, V. Falster, M.B. Doest, M.M. Ribo, E.R. Eiriksson, D.B. Pedersen, J.R. Frisvad, Microstructure control in 3d printing with digital light processing, Comput. Graph. Forum 39 (1) (2020) 347–359.

[22] S. Moylan, A. Cooke, K. Jurrens, J. Slotwinski, M.A. Dommez, A Review of Test Artifacts for Additive Manufacturing: NIST, US Department of Commerce, National Institute of Standards and Technology, 2012.

[23] A. Bazhanov, D. Yudin, V. Porkhalo, Development of modular control software for construction 3d-printer, IOP Conference Series: Materials Science and Engineering,

Vol. 327, IOP Publishing, 2018, p. 22011.
[24] H.-S. Byun, K.H. Lee, Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making, Robot. Comput.-Integr. Manuf. 22 (1) (2006) 69–80.
[25] R. Hu, H. Li, H. Zhang, D. Cohen-Or, Approximate pyramidal shape decomposition, ACM Trans. Graph. 33 (6) (2014) 213:1–213:12.
[26] W. Gao, Y. Zhang, D.C. Nazzetta, K. Ramani, R.J. Cipra, Revomaker: Enabling multi-directional and functionally-embedded 3d printing using a rotational cuboidal platform, Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (2015) 437–446.
[27] Z. Zeng, X. Yu, K. He, W. Huang, Z. Fu, Iterated tabu search and variable neighborhood descent for packing unequal circles into a circular container, Eur. J. Oper. Res. 250 (2) (2016) 615–627.
[28] K. Sugihara, M. Sawai, H. Sano, D.-S. Kim, D. Kim, Disk packing for the estimation of the size of a wire bundle, Jpn. J. Ind. Appl. Math. 21 (3) (2004) 259–278.
[29] M. Lee, K. Sugihara, D.-S. Kim, Topology-oriented incremental algorithm for the robust construction of the Voronoi diagrams of disks, ACM Trans. Math. Softw. 43 (2) (2016) 14:1-14:23.
[30] M. Lee, Topology-Oriented Incremental Algorithm for the Robust Construction of the voronoi Diagram of Disks and Balls (Ph.D. thesis), Graduate School of Hanyang University (August), 2019.
[31] M. Lee, K. Sugihara, D.-S. Kim, Robust construction of the Voronoi diagram of spherical balls in the three-dimensional space, Appl. Math. Comput. (2020) (Unpublished results).
[32] K. Sugihara, M. Iri, Construction of the Voronoi diagram for “one million” generators in single-precision arithmetic, Proc. IEEE 80 (9) (1992) 1471–1484.
[33] Slic3r, g-Code Generator for 3d Printers, (2018) https://slic3r.org/.
[34] Ultimaker Cura Software, (2019) https://ultimaker.com/software/ultimaker-cura.
[35] R. Comminal, M.P. Serdeczny, D.B. Pedersen, J. Spangenberg, Motion planning and numerical simulation of material deposition at corners in extrusion additive manufacturing, Addit. Manuf. 29 (2019) 100753.
[36] D.-S. Kim, I.-K. Hwang, B.-J. Park, Representing the Voronoi diagram of a simple polygon using rational quadratic Bézier curves, Comput.-Aided Des. 27 (8) (1995) 605–614.
[37] D.-S. Kim, Polygon offsetting using a Voronoi diagram and two stacks, Comput.-Aided Des. 30 (14) (1998) 1069–1076.
[38] ANSYS Academic Structural Release 19.2, (2019) https://www.ansys.com/products/structures.
[39] ANSYS Discovery Space Claim Release 19.2, (2019) https://www.ansys.com/en-gb/products/3d-design/ansys-spaceclaim.
[40] 3DWOX Desktop Software, (2019) https://3dprinter.sindoh.com/.
[41] Nc viewer. https://ncviewer.com/.
[42] P. Kulkarni, D. Dutta, Deposition strategies and resulting part stiffnesses in fused deposition modeling, J. Manuf. Sci. Eng. 121 (1) (1999) 93–103.
[43] Y. an Jin, H. Li, Y. He, J. zhong Fu, Quantitative analysis of surface profile in fused deposition modelling, Addit. Manuf. 8 (2015) 142–148.
[44] B.H. Kim, B.K. Choi, Machining efficiency comparison direction-parallel tool path with contour-parallel tool path, Comput.-Aided Des. 34 (2) (2002) 89–95.
[45] A. Telea, A. Jalba, Voxel-based assessment of printability of 3d shapes, International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing (2011) 393–404.
[46] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, W. Wang, Medial axis tree – an internal supporting structure for 3d printing, Comput. Aided Geometr. Des. 35–36 (2015) 149–162.
[47] P. Musialski, T. Auzinger, M. Birsak, M. Wimmer, L. Kobbelt, Reduced-order shape optimization using offset surfaces, ACM Trans. Graph. 34 (4) (2015) 1–9.
[48] R. Prévost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung, Make it stand: balancing shapes for 3d fabrication, ACM Trans. Graph. 32 (4) (2013) 81:1–81:10.
[49] S. Hornus, S. Lefebvre, Iterative Carving for Self-Supporting 3d Printed Cavities, Research Report RR-9083, Inria Nancy – Grand Est (July), 2017.
[50] J. Martínez, S. Hornus, H. Song, S. Lefebvre, Polyhedral Voronoi diagrams for additive manufacturing, ACM Trans. Graph. 37 (4) (2018) 129:1–129:15.
[51] Y. Zhou, H. Lu, Q. Ren, Y. Li, Generation of a tree-like support structure for fused deposition modelling based on the l-system and an octree, Graph. Models 101 (2019) 8–16.
[52] N. Zhang, L.-C. Zhang, Y. Chen, Y.-S. Shi, Local barycenter based efficient tree-support generation for 3d printing, Comput.-Aided Des. 115 (2019) 277–292.
[53] T. Tricard, F. Claux, S. Lefebvre, Ribbed support vaults for 3d printing of hollowed objects, Comput. Graph. Forum (2019) 1–13.
[54] B. Vaissier, J.-P. Pernot, L. Chougrani, P. Véron, Genetic-algorithm based framework for lattice support structure optimization in additive manufacturing, Comput.-Aided Des. 110 (2019) 11–23.
[55] E. Ulu, J. McCann, L.B. Kara, Structural design using laplacian shells, Eurographics Symposium on Geometry Processing 2019, Vol. 38 (2019).
[56] P. Kulkarni, A. Marsan, D. Dutta, A review of process planning techniques in layered manufacturing, Rapid Prototyp. J. 6 (1) (2000) 18–35.
[57] F. Rengier, A. Mehndiratta, H. von Tengg-Kobligk, C.M. Zechmann, R. Unterhinninghofen, H.-U. Kauczor, F.L. Giesel, 3d printing based on imaging data: review of medical applications, Int. J. Comput. Assist. Radiol. Surg. 5 (4) (2010) 335–341.
[58] A.T. Gaynor, J.K. Guest, Topology optimization considering overhang constraints: eliminating sacrificial support material in additive manufacturing through design, Struct. Multidisc. Optim. 54 (5) (2016) 1157–1172.
[59] T.E. Johnson, A.T. Gaynor, Three-dimensional projection-based topology optimization for prescribed-angle self-supporting additively manufactured structures, Addit. Manuf. 24 (2018) 667–686.
[60] J. Wang, J. Dai, K.-S. Li, J. Wang, M. Wei, M. Pang, Cost-effective printing of 3d objects with self-supporting property, Visual Comput. 35 (5) (2019) 639–651.
[61] J. Jiang, X. Xu, J. Stringer, Optimization of process planning for reducing material waste in extrusion based additive manufacuring, Robot. Comput.-Integr. Manuf. 59 (2019) 317–325.
[62] J.D. Deaton, R.V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, Struct. Multidisc. Optim. 49 (1) (2014) 1–38.
[63] B.K. Choi, R.B. Jerard, Sculptured Surface Machining: Theory and Applications, Springer, 2012.
[64] G. Zhao, C. Zhou, D. Lin, Tool path planning for directional freezing-based three-dimensional printing of nanomaterials, J. Micro Nano-Manuf. 6 (1) (2018).
[65] Y. an Jin, Y. He, J.-Z. Fu, W. feng Gan, Z. wei Lin, Optimization of tool-path generation for material extrusion-based additive manufacturing technology, Addit. Manuf. 1–4 (2014) 32–47.
[66] N. Ahsan, B. Khoda, AM optimization framework for part and process attributes through geometric analysis, Addit. Manuf. 11 (2016) 85–96.
[67] G.Q. Jin, W.D. Li, L. Gao, An adaptive process planning approach of rapid prototyping and manufacturing, Robot. Comput.-Integr. Manuf. 29 (1) (2013) 23–38.
[68] G.Q. Jin, W.D. Li, L. Gao, K. Popplewell, A hybrid and adaptive tool-path generation approach of rapid prototyping and manufacturing for biomedical models, Comput. Ind. 64 (3) (2013) 336–349.
[69] M. Held, J. Klosowski, J. Mitchell, Evaluation of collision detection methods for virdual reality fly-through, Proceedings of the Canadian Conference on Computational Geometry (1995).
[70] M. Held, S. Huber, Topology-oriented incremental computation of Voronoi diagrams of circular arcs and straight-line segments, Comput.-Aided Des. 41 (5) (2009) 327–338.
[71] Y. Yang, H.T. Loh, J.Y.H. Fuh, Y.G. Wang, Equidistant path generation for improving scanning efficiency in layered manufacturing, Rapid Prototyp. J. 8 (1) (2002) 30–37.
[72] Y. Nilsiam, P. Sanders, J.M. Pearce, Slicer and process improvements for open-source GMAW-based metal 3-d printing, Addit. Manuf. 18 (2017) 110–120.
[73] A. Roschli, K.T. Gaul, A.M. Boulger, B.K. Post, P.C. Chesser, Designing for big area additive manufacturing, Addit. Manuf. 25 (2019) 275–285.
[74] D. Ding, Z. Pan, D. Cuiuri, HuijunLi, A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures, Robot. Comput.-Integr. Manuf. 34 (2015) 8–19.
[75] A. Papacharalampopoulos, H. Bikas, P. Stavropoulos, Path planning for the infill of 3d printed parts utilizing hilbert curves, Proc. Manuf. 21 (2018) 757–764.
[76] S.H. Nair, A. Sinha, L. Vachhani, Hilbert’s space-filling curve for regions with holes, 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (2017).
[77] N. Kumar, S. Shaikh, P.K. Jain, P. Tandon, Effect of fractal curve based toolpath on part strength in fused deposition modelling, Int. J. Rapid Manuf. 5 (2) (2015) 186–198.
[78] S. Kapil, P. Joshi, H.V. Yagani, D. Rana, P.M. Kulkarni, R. Kumar, K. Karunakaran, Optimal space filling for additive manufacturing, Rapid Prototyp. J. 22 (4) (2016) 660–675.
[79] S. Shaikh, N. Kumar, P.K. Jain, P. Tandon, Hilbert curve based toolpath for fdm process, CAD/CAM, Robot. Fact. Future (2016) 751–759.
[80] Y. Zhang, A. Bernard, R. Harik, G. Fadel, A new method for single-layer-part nesting in additive manufacturing, Rapid Prototyp. J. 24 (5) (2018) 840–854.
[81] V. Griffiths, J.P. Scanlan, M.H. Eres, A. Martinez-Sykora, P. Chinchapatnam, Cost-driven build orientation and bin packing of parts in selective laser melting (SLM), Eur. J. Oper. Res. 273 (1) (2019) 334–352.
[82] J. Zhang, X. Yao, Y. Li, Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing, Int. J. Prod. Res. (2019) 1–20.
[83] X. Chen, C. Wang, X. Ye, Y. Xiao, S. Huang, Direct slicing from powershape models for rapid prototyping, Int. J. Adv. Manuf. Technol. 17 (7) (2001) 543–547.
[84] Y. Shi, X. Chen, D. Cai, S. Huang, Application software system based on direct slicing for rapid prototyping, Int. J. Prod. Res. 42 (11) (2004) 2227–2242.
[85] W. Cao, Y. Miyamoto, Direct slicing from autocad solid models for rapid prototyping, Int. J. Adv. Manuf. Technol. 21 (10–11) (2003) 739–742.
[86] F. Javidrad, A.R. Pourmoayed, Contour curve reconstruction from cloud data for rapid prototyping, Robot. Comput.-Integr. 27 (2) (2011) 397–404.
[87] W. Ma, W.-C. But, P. He, Nurbs-based adaptive slicing for efficient rapid prototyping, Comput.-Aided Des. 36 (13) (2004) 1309–1325.
[88] B. Starly, A. Lau, W.L.W. Sun, T. Bradbury, Direct slicing of step based nurbs models for layered manufacturing, Comput.-Aided Des. 37 (4) (2005) 387–397.
[89] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd ed, John Wiley & Sons, Chichester, 1999.
[90] D.-S. Kim, D. Kim, K. Sugihara, Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology, Comput. Aided Geometr. Des. 18 (2001) 541–562.
[91] D.-S. Kim, D. Kim, K. Sugihara, Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry, Comput. Aided Geometr. Des. 18 (2001) 563–585.
[92] L. Jin, D. Kim, L. Mu, D.-S. Kim, S.-M. Hu, A sweepline algorithm for Euclidean Voronoi diagram of circles, Comput.-Aided Des. 38 (3) (2006) 260–272.
[93] I.Z. Emiris, G.M. Tzoumas, Algebraic study of the Apollonius circle of three ellipses, EuroCG (2005) 147–150.
[94] H. Alt, O. Cheong, A. Vigneron, The Voronoi diagram of curved objects, Discr. Comput. Geometry 34 (3) (2005) 439–453.