

A Support-free Infill Structure Based on Layer Construction for 3D Printing

Wenpeng Xu, Yi Liu, Menglin Yu, Dongxiao Wang, Shouming Hou, Bo Li, Weiming Wang*, and Ligang Liu,

Abstract—The design of the light-weight infill structure is a hot research topic in additive manufacturing. In recent years, various infill structures have been proposed to reduce the amount of printing material. However, 3D models filled with them may have very different structural performances under different loading conditions. In addition, most of them are not self-supporting. To mitigate these issues, a novel light-weight infill structure based on the layer construction is proposed in this paper. The layers of the proposed infill structure continuously and periodically transform between triangles and hexagons. The geometries of two adjacent layers are controlled to be self-supporting for different 3D printing technologies. The machine code (Gcode) of the filled 3D model is generated in the construction of the infill structure for 3D printers. That means 3D models filled with the proposed infill structure do not need an extra slicing process before printing, which is time consuming in some cases. Structural simulations and physical experiments demonstrate that our infill structure has comparable structural performance under different loading conditions. Furthermore, the relationship between the structural stiffness and the parameters of the infill structure is investigated, which will be helpful for non-professional users.

Index Terms—Light-weight structure, 3D printing, Self-supporting, Structural performance, Machine code.

1 INTRODUCTION

THE advance of 3D printing (or Additive Manufacturing (AM)) technologies enables us to fabricate arbitrary complex 3D objects. At present, a large number of materials have been used in 3D printing, such as plastic, metal and so on. However, the cost of some materials is still high, especially for the metal material. Therefore, reducing printing materials is a hot and active research topic. Hollowing is a simple and straightforward way to reduce printing materials, which has been integrated into commercial printing softwares. The user only needs to give a hollowing thickness, and then a shell model with a specific thickness is generated. In addition, several simple infill structures, such as the triangle prism, the hexagonal prism, and the quadrangular prism (see Figure 1), are provided in commercial printing software to reduce printing material while maintaining a certain structural stiffness for the extrusion-based 3D printing technologies, such as Fused Deposition Modeling (FDM). However, it is difficult for non-professional users to select an appropriate fill ratio between material usage and structural stiffness. Therefore, it is necessary to propose automatic methods to reduce the material usage while maintaining a certain structural stiffness.

To handle the above issues, a lot of light-weight infill structures [1] have been proposed, such as the Skin-frame structure [2], Honeycomb-cell structure [3], Media-

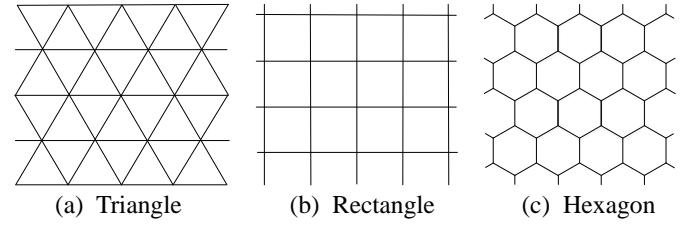


Fig. 1. The illustration of three simple infill patterns commonly used in commercial 3D printing softwares.

axis structure [4], Polyhedral Voronoi [5], Crossfill [6] and so on. Unfortunately, most of them are not self-supporting [2], [3], [4]. As a result, extra support structures are added during the printing, which are difficult to be completely removed. Some recent 3D printers can print two or more materials. It is possible to print soluble support structures which are easy to remove. However, the soluble support structures can only be removed and cleaned for the outer space or interior with holes. In practice, for the models with complicated closed holes, such as the Honeycomb model [3], it is very difficult to clean the internal soluble support structure. In order to avoid the above issues, extensive self-supporting infill structures have been proposed recently, such as the Rhombic [7], Support-free hollowing [8], Elliptic hollowing [9] and so on. The term self-supporting has been used in different contexts. In architectural geometry it refers to a structure (i.e., an arrangement of blocks such as bricks, stones) which is in a static equilibrium configuration [10], [11], [12]. In this work, the self-supporting is a geometric property concerning overhanging angles. For an interior surface, it is self-supporting if and only if its faces are self-supporting. That is the following inequality is satisfied for all faces:

$$(\mathbf{n}_{f_j} \cdot \mathbf{d}) \geq \cos\left(\frac{\pi}{2} + \alpha\right), \quad (1)$$

- W. Xu, Y. Liu, D. Wang, S. Hou are with the School of Computer Science Technology, Henan Polytechnic University, Jiaozuo, China.
- M. Yu is with the School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, China.
- B. Li is with the School of Mathematics and Information Science, Nanchang Hangkong University, Nanchang, China.
- W. Wang is with the Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province, School of Mathematical Sciences, Dalian University of Technology, Dalian, China.
- L. Liu is with the School of Mathematical Sciences, University of Science and Technology of China, Hefei, China.
- Corresponding Author: W. Wang (Email: wwmdlut@dlut.edu.cn).

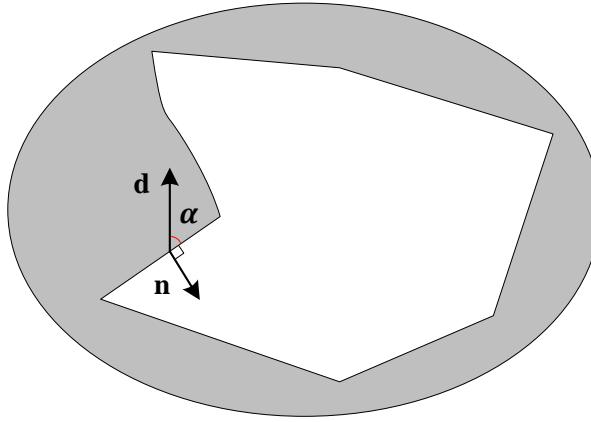


Fig. 2. The notations of the self-supporting for the faces.

where \mathbf{n}_{f_j} is the normal of the j^{th} face, \mathbf{d} is the printing direction, (\cdot) is the inner product between two vectors, and $\alpha \in [0, \frac{\pi}{2}]$ is the maximal self-supporting angle (SSA) or the angle between \mathbf{d} and the face (see Figure 2).

Although many infill structures are self-supporting themselves, such as [7], [8], [9], some of them may not be self-supporting any more after intersecting with the objects which have overhanging surfaces. As a result, the objects filled with them may not be self-supporting. In addition, most infill structures have very different structural performances under different loading conditions.

To solve the above issues, a novel infill structure is proposed in this work. For our infill structure, its layers continuously and periodically transform between triangles and hexagons from bottom to top. Therefore, several good properties of triangle and hexagon are naturally integrated into our infill structure, such as stability and light-weight, etc. Our infill structure has comparable structural performance under different loading conditions, which will be proved by the simulating and physical tests in Section (5). In addition, the geometry of the infill structure is controlled to be self-supporting for different kinds of 3D printing technologies (see Figure 5 and Section (4.3)). To achieve the support-free printing of a filled object with overhanging surfaces, a subdivision strategy is proposed in Section (5.4).

The main contributions of this work are as follows:

- A novel light-weight infill structure is proposed which is geometrically support-free and has comparable structural performance under different loading conditions.
- The machine code for 3D printers can be automatically generated in the construction of our infill structure, which avoids an extra slicing process before printing.
- The relationship between the maximal Von Mises Stress (abbreviated as MVMS in the following) and the parameters of the infill structure is studied, which will be helpful for non-professional users.

The rest of this paper is organized as follows. The related articles will be reviewed in Section (2). The pipeline of our method will be described in Section (3). After that, we will introduce the details of the structural construction in Section (4.1), the self-supporting property in Section (4.2), and

the reconstruction of 3D infill structure in Section (4.3). Some representative results are shown in Section (5) to validate the effectiveness of our approach, which are followed by the conclusions and ideas about the future works in Section (6).

2 RELATED WORK

With the development of additive manufacturing technologies in recent years, a large number of methods have been proposed to deal with geometric and structural problems in AM [13]. In this work, we mainly focus on the approaches closely related to our method, that is the light-weight infill structures, the self-supporting structures, and the structural analysis and optimization for 3D printing. The interested readers can refer to [14], [15], [16] for more algorithms.

2.1 Light-weight infill structures

Many studies have been devoted to design the light-weight structures to reduce the amount of material while maintaining certain structural performance for AM, such as the structural soundness and stability, etc. A sparse optimization method was proposed in [2] to generate the skin-frame structure in 3D models subject to various constraints, such as the stiffness, stability, and printability, for the purpose of reducing the material usage. Similarly, Zhang et al. [4] used the medial axis instead of the frame structure, which can naturally transfer the external loads from different directions to the inner core structure. The honeycomb-cell structure which is known to be of minimal material usage while providing strength in tension, was used to divide the interior of the model [3] and [17]. The porous structures are widely seen in nature, which are light-weight and exhibit strong structural properties. Therefore, Wu et al. [18] presented a method to generate bone-like porous structures with topology optimization. In [5] and [6], two different microstructures were proposed to design structures with specific elastic properties. However, the infill structures proposed in [2], [3], [4], [17], [18] are not self-supporting, which may introduce the need for more supporting structures during printing. The supporting structures are always difficult to be completely removed.

2.2 Self-supporting structures

An intrinsic way to avoid internal support structure is to generate a self-supporting interior surface during the hollowing process. Recently, Reiner and Lefebvre [19] proposed an interactive modeling tool to design self-supporting models. Wu et al. [7] introduced the rhombic structure as a special self-supporting infill structure for 3D printing. The rhombic structure is adaptively refined according to the analysis of physical properties, such as the center of mass and the stress distribution. The hollowing operator presented in [20] generated even sparser structures. Wang et al. [8] proposed a sparsity optimization framework with overhanging constraints to generate support-free frame structures. Yang et al. [21] presented a simple hollow-to-fill algorithm to generate support-free inner surfaces with shape optimization. Xie and Chen [22] recently proposed a method to generate support-free interior cavities. However,

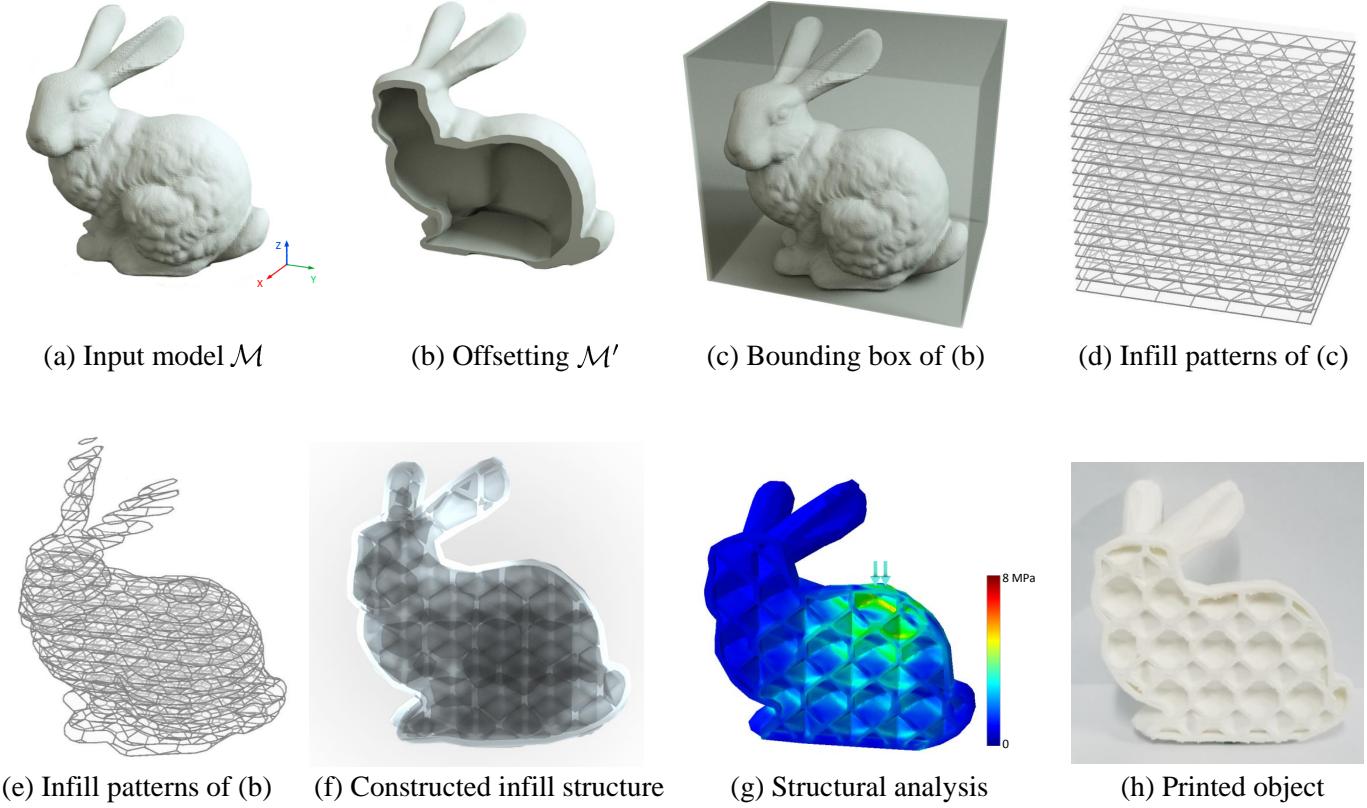


Fig. 3. The pipeline of our algorithm. The edge length r and thickness t of the reconstructed infill structure in (f) are 4mm and 0.7mm, respectively. The force in (g) is 200N and the corresponding MVMS is 7.887MPa. (f) is rendered with transparency to clearly see the interior infill structure. (h) shows the printed object filled with our infill structure.

their method is voxel-based and thus results in staircase-like inner surfaces. Support-free elliptic hollowing method was proposed by Lee et al. [9]. However, most of the above infill structures may result in very different structural performances under different loading conditions.

2.3 Structural strength analysis and optimization

Currently, the acquisition of 3D models and manufacturing are separated. As a result, many 3D models are difficult to print or transport after printing since structural analysis is missing. Therefore, it is important to provide convenient methods to detect and fix structural issues before printing. Telea and Jalba [23] proposed a method to detect structural weak regions, and a set of geometric rules and voxel-based representation were used to make sure the printability of 3D models. Umetani and Schmidt [24] proposed a cross-section based structural analysis method. In their work, structural stiffness was considered in the manufacturing direction optimization. Zhou et al. [25] presented a method to identify structural problems of objects without specific assumptions on loads and manual load setup. However, the above algorithms only detect the structural problematic regions without fixing them.

In [26], the authors first detected the structural weak regions with tetrahedron based FEM method. Then a set of operations were applied to improve the structural stiffness, such as hollowing, thickening, and strut insertion. Xie et al. [27] presented a fabrication-aware shape editing system.

3D models were iteratively edited according to the feedback from the structural analysis with FEM. Based on the cross sectional structural analysis, Xu et al. [28] used the skeleton of the 3D model to enhance its thin parts, and Wang et al. [29] proposed an optimization system to simultaneously hollow the structural strong regions and enhance the structural weak regions.

In contrast, we propose a novel support-free infill structure which has comparable structural performance under different loading conditions. Similar to Crossfill [6], Cubic and Octet [30], 3Dhoney [31], and Polyhedral Voronoi [5], the machine code is automatically generated in the construction of the infill structure, which largely reduces the time consumed in the preprocessing before printing. The relationship between MVMS and the parameters of the infill structure is investigated. Users can easily generate infill structures according to their requirements.

3 ALGORITHM OVERVIEW

In Figure 3, we show the pipeline of our algorithm. For a given 3D model M (see Figure 3 (a)), we first generate its offsetting version M' with [32] to guarantee the printability of the final object (see Figure 3 (b)). Then, an axis-aligned boundingbox $B_{M'}$ of M' is calculated (see Figure 3 (c)). And the layers of the infill structure are originally constructed in $B_{M'}$ (see Figure 3 (d)). We intersect the layers of M' with the corresponding layers of the filled $B_{M'}$ to generate the layers of the infill structure for M' (see

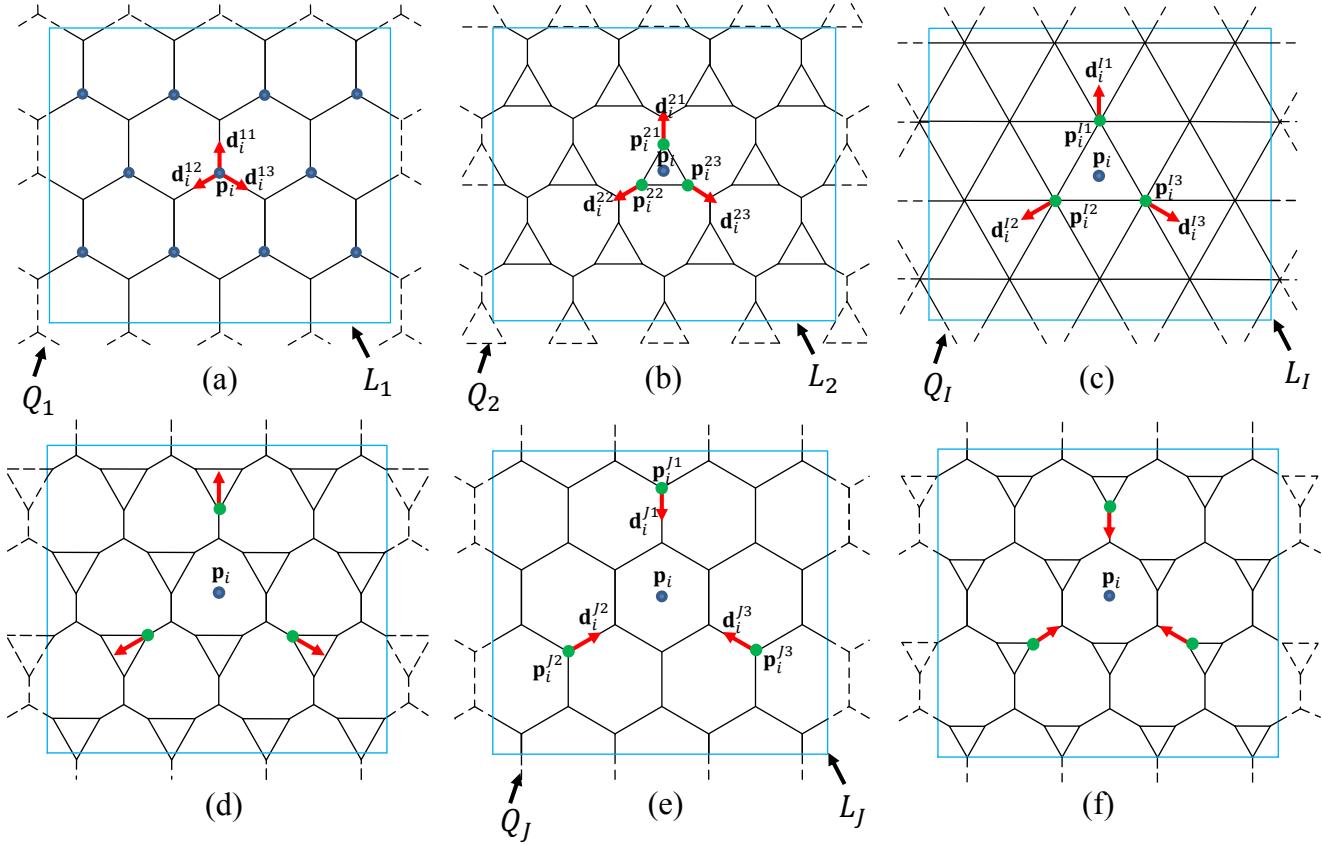


Fig. 4. Six layers are shown to demonstrate the construction of the layers of the proposed support-free infill structure. All blue nodes in (a) are equivalent, which have the same movement rule. We use p_i as an example to illustrate the movement of the associated nodes. The arrows indicate the moving directions of the corresponding nodes.

Figure 3 (e)). Further on, a strip triangulation algorithm [33] is applied to construct the 3D infill structure of \mathcal{M} , by connecting the contours in neighboring layers (see Figure 3 (f)). Finally, structural performance of the final model is analyzed (see Figure 3 (g)) and the model is fabricated (see Figure 3 (h)).

4 CONSTRUCTION OF INFILL STRUCTURE

4.1 Layers construction for infill structure

Similar to the layer upon layer 3D printing technologies, our infill structure is constructed in the same way. To do so, we slice $B_{\mathcal{M}'}$ into layers along the z -axis (the printing direction) with a given slicing thickness h . To simplify the following description, we project all layers of $B_{\mathcal{M}'}$ onto the printing platform, that is the $X - Y$ plane, which is represented as $\mathcal{L} = \{L_1, L_2, \dots, L_k\}$, where k is the number of layers. All layers in \mathcal{L} are 2D rectangles which will be filled with patterns to form the layers of our infill structure in $B_{\mathcal{M}'}$.

The construction starts from the first layer L_1 (the blue rectangle in Figure 4 (a)) in \mathcal{L} . We first completely fill L_1 with a hexagonal pattern Q_1 (the black pattern in (a)). Q_1 is composed of a set of nodes $P = \{p_1, p_2, \dots, p_n\}$ and a set of edges $E = \{e_1, e_2, \dots, e_m\}$. All edges have the same length r which can be given by users to determine the fill ratio. Each edge e_i is assigned a value t_i indicating the thickness of this edge, which can be adjusted to achieve

different structural performances. In our method, all edges have the same thickness t which also affects the fill ratio of the final model. The edge thickness will not be considered in the construction of the layers. However, they will be considered in the reconstruction of the final 3D infill structure in Section (4.2).

To construct the other patterns from Q_1 , we first randomly select a node p_i from P , and build a set of nodes S_p starting from p_i , where any two nodes in S_p cannot share the same edge (see the blue nodes in Figure 4 (a)). As we can see from Figure 4 (a), p_i is connected with three edges which are denoted as e_i^1, e_i^2 , and e_i^3 . Then, we construct a node-direction pair set $S_p^d = \{(p_i^{1w}, d_i^{1w}), (p_i^{2w}, d_i^{2w}), (p_i^{3w}, d_i^{3w}) \mid p_i \in S_p\}$. Here p_i^{1w}, p_i^{2w} , and p_i^{3w} are the same as p_i on Q_1 , and d_i^{1w}, d_i^{2w} , and d_i^{3w} are the normalized moving directions of the corresponding nodes, which are consistent with e_i^1, e_i^2 , and e_i^3 , respectively (see the red arrows in Figure 4 (a)). p_i^{jw} means the w^{th} node on the j^{th} layer corresponds to the node p_i in S_p . We first let all nodes in S_p^d move to the other end of the corresponding edges along their moving directions. The moving step size for all nodes in S_p^d is represented as s . Then Q_2 is constructed by moving all nodes in S_p^d a distance s along their moving directions (see Figure 4 (b)). The coordinates of the moving nodes on Q_2 are calculated as:

$$p_i^{2w} = p_i^{1w} + s d_i^{1w}, \quad w = 1, 2, 3. \quad (2)$$

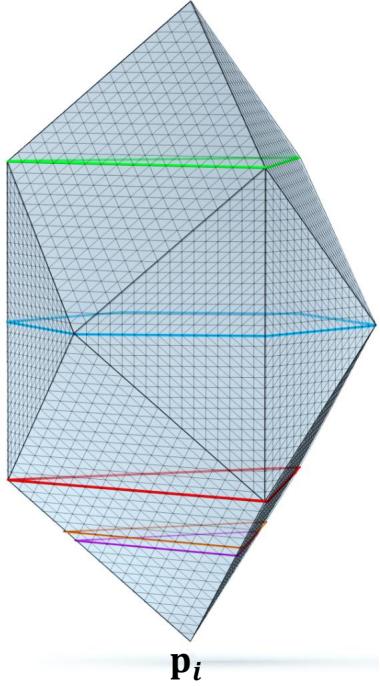


Fig. 5. The unit cell of our infill structure reconstructed from cross-sections. p_i is the starting node. The blue polygon is a hexagon cross-section. The red and green polygons are two equilateral triangle cross-sections. The orange and purple polygons are two neighboring cross-sections.

Here we should notice that the nodes on Q_1 , which are not selected into S_p , are fixed, and their connections are updated according to the associated moving nodes in the construction of the new patterns.

We then update the nodes in S_p^d with Equation (2), while the moving directions remain the same. The infill pattern Q_3 for the layer L_3 is constructed by moving all nodes in S_p^d along their moving directions a distance s , and S_p^d is updated again. This procedure is recursively implemented until all nodes in S_p^d reach the other end of the corresponding edges at a certain layer L_J . The current layer is filled with equilateral triangles and the nodes in S_p become the centers of the corresponding triangles (see Figure 4 (c)). The edge length of the equilateral triangle is $\sqrt{3}r$. Further on, all moving nodes continuously move forward along their moving directions (see Figure 4 (d)), until three of them meet together. At this time, the current layer L_J is filled with hexagons where all nodes in S_p become the centers of the hexagons in Figure 4 (e). After that, all nodes in S_p^d move backward to their corresponding nodes in S_p to generate the infill patterns for the other layers of \mathcal{L} (see Figure 4 (f)). At a certain layer, three moving nodes meet at one node in S_p^d (see Figure 4 (a)). So far, we have finished constructing the patterns of all layers of the unit cells from the nodes in S_p . The above procedure is recursively implemented until all layers in \mathcal{L} are filled with patterns. Finally, the layers of $B_{\mathcal{M}'}$, denoted as \mathcal{L}' , are generated by the intersection of the layers in \mathcal{L} and the corresponding infill patterns (see Figure 3 (d)). The construction of more layers in \mathcal{L} can be viewed in the accompanying video.

4.2 3D infill structure reconstruction

In this section, we reconstruct the 3D infill structure from the layers calculated in Section (4.1). A strip triangulation algorithm [33] can be adopted to construct the 3D infill structure in $B_{\mathcal{M}'}$, by connecting the contours between neighboring layers in \mathcal{L}' . In Figure 5, we show a unit cell of our infill structure reconstructed from the layers generated from p_i (see Figure 4 (a)). The red triangle corresponds to the layer in Figure 4 (c), and the blue hexagon corresponds to the layer in Figure 4 (e).

The final 3D infill structure of \mathcal{M} is generated by intersecting \mathcal{M}' with the constructed 3D infill structure in $B_{\mathcal{M}'}$. However, the intersection of two 3D models is a difficult task as geometrical and topological problems. Instead of 3D intersection, we prefer to calculate the intersections between the layers in \mathcal{L}' and the corresponding layers of \mathcal{M}' . Specifically, we first slice \mathcal{M}' along the z -axis into layers with thickness h . Then, we intersect the layers in \mathcal{L}' with the corresponding layers of \mathcal{M}' to generate the layers of the infill structure in \mathcal{M} , which is denoted as $\tilde{\mathcal{L}}$ (see Figure 3 (e)). Before constructing the 3D infill structure from layers, we offset each edge along its normal and anti-normal to generate an edge thickness t . At last, the strip triangulation algorithm [33] is applied on the cross-sections of all layers to generate the 3D infill structure, where a cross-section is a closed polygon and one layer may contain more than one cross-section (see Figure 3 (f)).

It is worth noticing that the 3D infill structure is constructed only for the structural simulation and analysis. To print the 3D models filled with our infill structure, we can directly generate the Gcode for 3D printer from $\tilde{\mathcal{L}}$. In this way, we don't need to spend extra time slicing complex 3D infill structure to generate the Gcode, which largely improves the efficiency of the 3D printing technologies. The algorithm described in Section 4.1 and the generation of the Gcode are implemented based on the Cura engine [34] which is a popular 3D printing engine. Our infill structure can be easily integrated into the commercial 3D printing softwares.

4.3 Self-supporting property

As described in Section (4.1), our infill structure is composed of unit cells. Therefore, if the unit cell is self-supporting, then the final infill structure will also be self-supporting. In this section, we only discuss the self-supporting property of the proposed infill structure itself. The self-supporting property of the object filled with our infill structure will be discussed in Section (5.4). As described in Section (1), the unit cell is self-supporting if and only if its faces are self-supporting, that is Equation (1) is satisfied for every face of the unit cell. The unit cell of our infill structure is constructed from the cross-sections generated in Section (4.1). To analyze the self-supporting property of a unit cell, we first project all its cross-sections onto the printing platform, that is the $X - Y$ plane. For two projected neighboring cross-sections, if the lower one is fully covered by the upper one, the surface constructed with them are self-supporting, for instance the orange and purple cross-sections shown in Figure 5. The reason is that the angle between the printing direction \mathbf{d} and the surface normal \mathbf{n} is smaller than $\frac{\pi}{2}$. That means

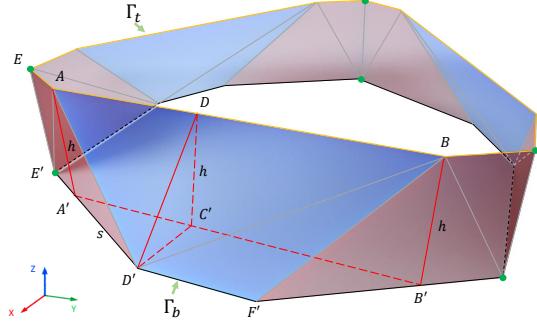


Fig. 6. The illustration of two adjacent cross-sections on a unit cell.

Equation (1) is automatically satisfied. Therefore, the faces constructed with the cross-sections between the bottom one and the blue one in Figure 5 are self-supporting. As a result, we only need to check the self-supporting property of the faces reconstructed with the cross-sections above the blue one. Here we should mention that the surface normal \mathbf{n} of the unit cell is inward as the unit cell will be inside of a 3D shape.

In Figure 6, we show two neighboring cross-sections Γ_b and Γ_t of a unit cell, which are above the blue cross-section shown in Figure 5. The faces (triangles) constructed with them are shown with the red and blue transparent colors. The grey segments are the edges cross Γ_b and Γ_t . When constructing Γ_t from Γ_b , the green nodes on Γ_b are fixed and the rest of nodes move along their moving directions a distance s . For instance, the node D' on Γ_b moves along the edge $D'E'$, which is its moving direction, a distance s to generate the node A' on Γ_b . Then the z coordinates of all fixed and moving nodes are added by h to generate the corresponding nodes on Γ_t . For instance, the node A is generated by adding the z coordinate of A' a value h . Therefore, A' is the projection node of A after projecting Γ_t onto the plane of Γ_b . The node B is generated in the same way. Therefore, it is easy to observe that the faces marked with red transparency are self-supporting, as they are perpendicular to the plane of Γ_b . Since the angle between the printing direction \mathbf{d} and the normals of the blue faces are the same, we use $\Delta ABD'$ as an example to analyze their self-supporting properties.

As shown in Figure 6, segments AA' and BB' are perpendicular to Γ_b as A' and B' are the projection nodes of A and B on Γ_b . Let $D'C' \perp A'B'$ and $DC' \parallel A'A$, then $DC' \perp D'C'$, thus $\angle D'DC'$ which is the angle between $\Delta ABD'$ and the printing direction (parallel to DC'), determines whether $\Delta ABD'$ is self-supporting or not. As a result, we get

$$\tan(\angle D'DC') = \frac{D'C'}{DC'}. \quad (3)$$

At Γ_b , the angles corresponding to the green nodes are 120° and the other angles are equal, then $\angle E'D'F' = 150^\circ$. As $D'C' \perp D'F'$, so $\angle D'A'C' = 30^\circ$. With $A'D' = s$ and $DC' = h$, we obtain $D'C' = \frac{s}{2}$ and

$$\tan(\angle D'DC') = \frac{s}{2h}. \quad (4)$$

For most 3D printing technologies, the maximal self-supporting angle is $\alpha = 45^\circ$, which means $s \leq 2h$ for our

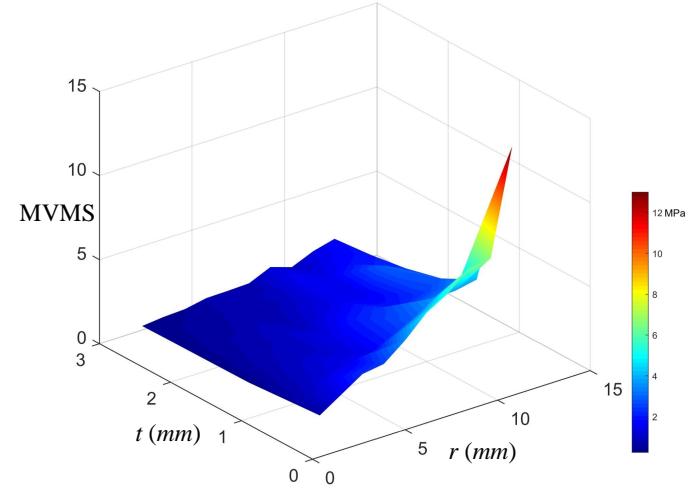


Fig. 7. The relationship among the edge length r , the edge thickness t , and MVMS.

infill structure. For the other 3D printing technologies, the maximal self-supporting angles can be different. However, we can easily select a suitable moving step size s in Equation (4) to achieve self-supporting properties for these 3D printing technologies. The infill structures generated with different $\angle D'DC'$ will be discussed in Section (4.4).

4.4 Algorithm discussions

As described in the previous sections, several parameters of our algorithm affect the shape and structural performance of the model filled by our infill structure, which are the edge length r , the edge thickness t , and the moving step size s . r and t directly affect the fill ratio and structural stiffness of the final objects. s directly affects the self-supporting property of the unit cell. Therefore, we will first discuss these parameters in this subsection. Then, we will discuss the influence of the position and orientation of the first pattern Q_1 on the final infill structure at the end of this subsection.

To analyze the effect of different r and t on the structural performance, we first uniformly sample r between 0 and 15mm and t between 0 and 3mm. Then we generate a set of (r, t) pairs which is represented as $\mathcal{Y} = \{(r, t) | 0 < r \leq 15, 0 < t \leq 3\}$. For each pair in \mathcal{Y} , a cube model is filled with the infill structure constructed by this parameter pair. Then the structural performance of this model is analyzed by considering a load on its top, and MVMS is recorded. In Figure 7, we show the functional relationship among r , t , and MVMS. Here s is equal to $2h$. With this function, users can easily select an appropriate infill structure according to their requirements.

Different 3D printing technologies may have different maximal self-supporting angle α . In our method, the moving step size s in Equation (4) can be easily changed to generate the infill structures with different face angles to adopt to different 3D printing technologies. In Figure 8, we show the infill structures generated with four different s , which are $3.46h$, $2h$, $1.15h$, and $0.27h$ from left to right, where h is the slicing thickness. As we can see from Figure 8, the smaller the s , the higher the unit cell. The reason is that

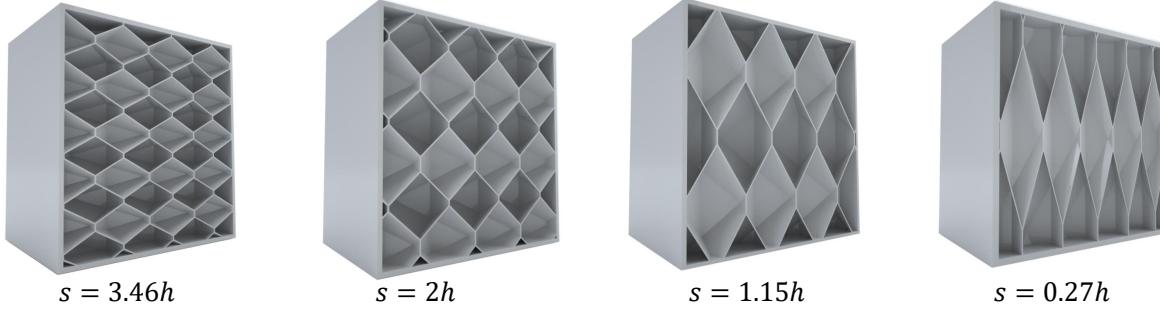


Fig. 8. The infill structures generated with different moving step sizes s which are $3.46h$, $2h$, $1.15h$, and $0.27h$ from left to right. The corresponding angles between the horizontal plane and the faces of the unit cells are 30° , 45° , 60° , and 75° , respectively. h is the slicing thickness which is a constant.

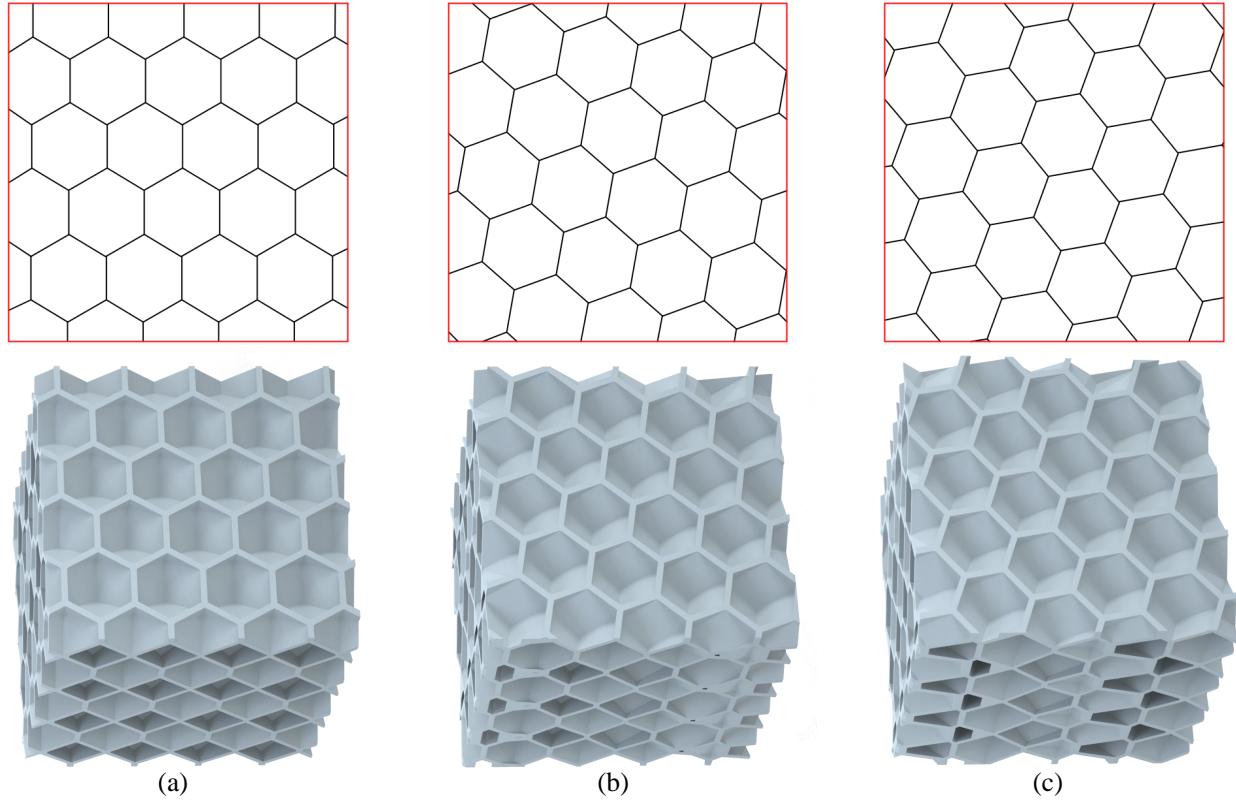


Fig. 9. Three different initial patterns Q_1 are shown in the first row. (a) is the same as that shown in Figure 4(a). (b) and (c) are generated by rotating (a) 10° and 20° clockwise, respectively. The corresponding 3D infill structures are shown in the second row.

more layers are needed to construct the unit cell when using a smaller s .

In the first row of Figure 9, Q_1 with three different orientations are shown. (a) is the same as Figure 4(a). (b) and (c) are generated by rotating (a) 10° and 20° clockwise, respectively. The constructed infill structures with the corresponding Q_1 are shown in the second row. As we can see, the infill structures have the same unit cell but with different orientations which are determined by the orientation of Q_1 . For the position of Q_1 , it only provides a translation of the final infill structure, and does not affect the shape and performance of the infill structure. Therefore, we can conclude that Q_1 can be placed in any position and orientation, which only affects the orientation of the final infill structure. The pattern shown in Figure 4 (a) is just for

the convenience of describing our algorithm.

5 EXPERIMENTAL RESULTS

Our algorithm was implemented with C/C++, and it was run on a PC with Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 32GB memory, and NVIDIA GeForce 920A GPU. The final structures are fabricated by an Ultimaker3 3D printer with 0.4mm print cores, and the slicing thickness is $h = 0.2\text{mm}$. The material is PLA plastic whose Young's modulus is 3000MPa and possion ratio is 0.35 . The infill line width is 0.4mm and the speed is 20mm/s . Since 45° is enough to achieve the self-supporting property for the Ultimaker printer, s in Equation (4) is set as 0.4mm . All simulation results shown in this work were calculated with the simulation software Abaqus [35].

TABLE 1

The statistics of the displacements (ΔL), the Young's modulus (E), and MVMS measured under 13 different directions for the filled sphere model shown in Figure 10. The numbers in the first row correspond to the direction index (DI) shown in Figure 10 (left). The vectors of these directions (DV) are listed in the second row. The pressure is 0.7kN.

DI	1	2	3	4	5	6	7	8	9	10	11	12	13
DV	(-1,0,0)	(0,-1,0)	(0,0,-1)	(-1,-1,-1)	(-1,-1,1)	(-1,1,-1)	(-1,1,1)	(-1,0,-1)	(-1,0,1)	(-1,-1,0)	(-1,1,0)	(0,-1,-1)	(0,-1,1)
ΔL (mm)	0.0798	0.0816	0.0834	0.0773	0.0771	0.0773	0.0784	0.0784	0.0784	0.0824	0.0826	0.0759	0.0767
E (MPa)	158.3	154.7	151.5	163.4	163.8	163.5	161.1	161.0	161.0	153.2	152.9	166.4	164.6
MVMS(MPa)	2.755	2.533	2.891	3.125	3.271	3.178	3.192	2.838	2.925	2.826	2.847	3.087	2.956

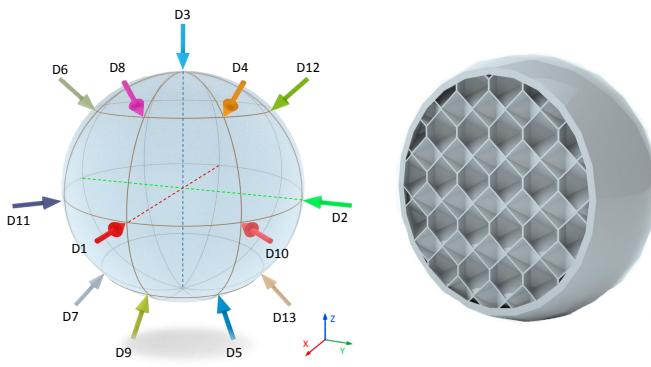


Fig. 10. Structural performance of a sphere model (left) filled with our infill structure (right) is evaluated in 13 loading directions, uniformly sampled on the sphere (see the arrows). The direction vectors are listed in Table 1. The infill structure generated with $r = 11.6\text{mm}$ and $t = 1.0\text{mm}$ is shown on the right. The model is cut to see the interior infill structure.

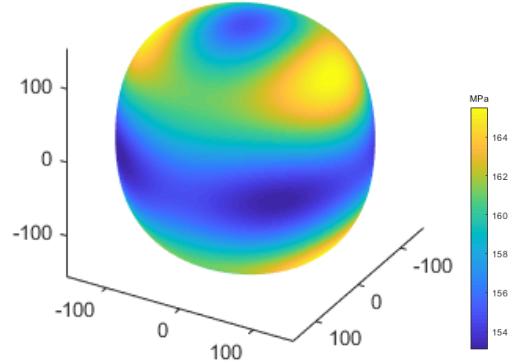


Fig. 11. The directional Young's modulus of the filled sphere model shown in Figure 10.

5.1 Structural performance evaluation

A sphere model is filled by our infill structure with $r = 11.6\text{mm}$ and $t = 1.0\text{mm}$, which is shown in Figure 10 (right). We uniformly sample 13 directions on the sphere indicated as arrows in Figure 10 (left). The index and direction vectors of the forces are listed at the first and second rows of Table 1, respectively. Then the displacements, the Young's modulus, and MVMS of the filled sphere are measured under the pressure (0.7kN) in these directions [36], which are listed at the last three rows of Table 1. From the last two rows we can see, our infill structure has very similar Young's modulus and MVMS in different loading directions.

In Figure 11, the directional Young's modulus is plotted, which is generated by interpolating the Young's modulus values listed in Table 1. The colormap indicates the values of the Young's modulus, where the blue regions have the smallest Young's modulus and the yellow regions have the largest Young's modulus. From Table 1 and Figure 11 we can clearly see, our infill structure has comparable structural performance in different loading directions.

5.2 Structural comparisons

In this section, we first compare our infill structure (see Figure 12 (c)) with two commonly used infill structures in 3D printing software, that is hexagonal prism (see Figure 12 (a)) and triangle prism (see Figure 12 (b)), through structural simulation. To conduct fair comparisons, all models are generated with the same fill ratio. In the first row, the stiffness of the models is tested under the longitudinal forces which are 200N. The MVMSs of (a) and (b) are smaller than (c) (2.647MPa) as hexagonal and triangle prisms are good at the transmission of the longitudinal forces. In the second row, the stiffness of these models is tested under the lateral forces which are 200N. We can observe that the MVMS of (c) (2.527MPa) is smaller than (a) (22.813MPa). Although the MVMSs of (b) and (c) are comparable, (c) is better than (b). From the above comparisons we can conclude, hexagonal and triangle prisms cause obviously different structural performances in longitudinal and lateral loading directions, while our infill structure has balanced structural performance.

In Figure 13, eight physical tests are conducted to evaluate the structural performances among the hexagonal prism [34] (a), the triangle prism [34] (b), Crossfill [6] (c), Polyhedral Voronoi [5] (d), Octet [30] (e), Cubic [30] (f), 3Dhoney [37] (g), and our method (h). All objects are printed with the Ultimaker 3D printer. The size of the objects is $50\text{mm} \times 50\text{mm} \times 50\text{mm}$ and the weight of each object is $16\text{g} \pm 0.1\text{g}$. The objects are pressed in the lateral direction. The first and third rows show the initial status without pressure. The second and fourth rows show the status under the test. As we can see, the hexagonal prism is almost crashed under a small pressure. The triangle prism is stiffer than 3Dhoney, but weaker than the other infill structures. Crossfill is better than Octet as 1.0021kN pressure generates 4.79mm displacement. Polyhedral Voronoi, Cubic, and our method have similar performance and outperform the other infill structures. However, our method generates the smallest displacement 4.27mm under the largest pressure 1.0077kN. In Figure 14, we plot the displacement-pressure

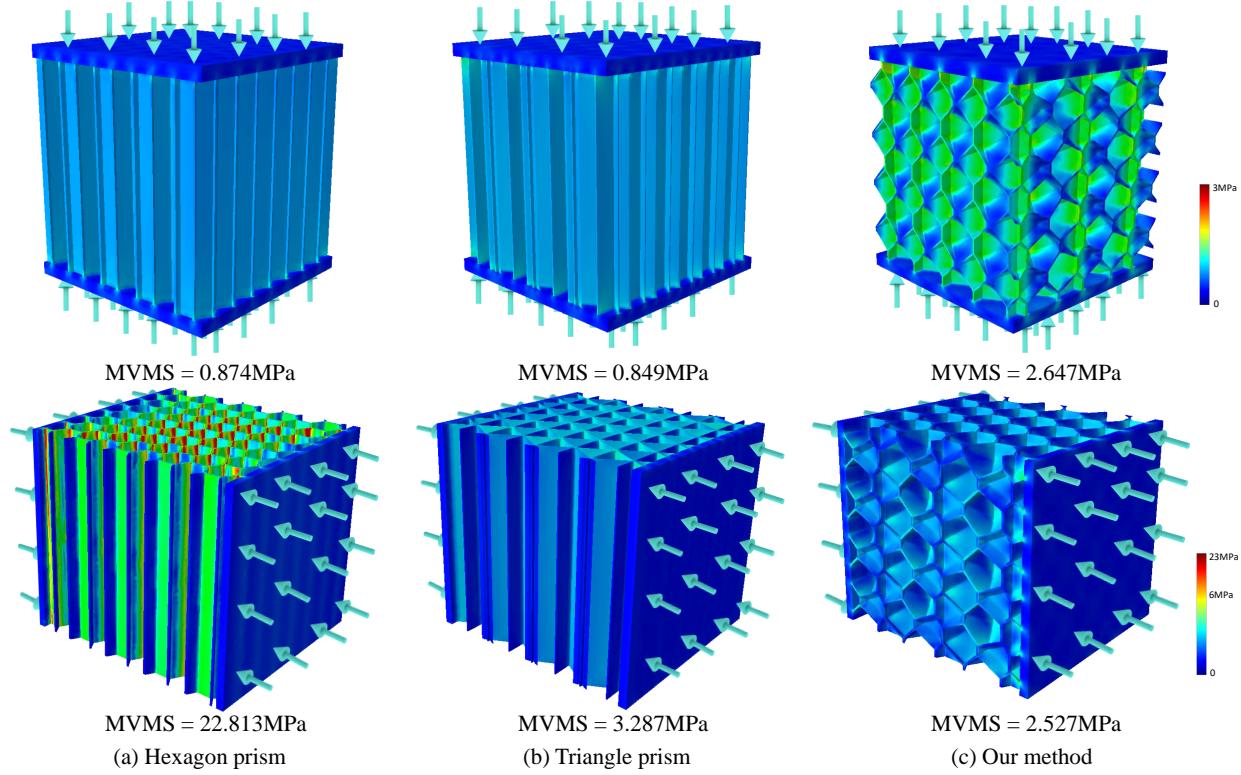


Fig. 12. Structural comparisons among the hexagonal prism (a), the triangle prism (b), and our infill structure (c) under the same fill ratio. Structural performances are evaluated by considering two different loading conditions: the longitudinal forces (the first row), and the lateral forces (the second row). The colormap shows the residual stress where the dark blue indicates the smallest stress and the dark red indicates the largest stress. MVMSs are listed at the bottom.

curves of the physical tests shown in Figure 13. The red curve represents our method. From these curves we can see, Polyhedral Voronoi, Crossfill, Cubic, and our method have very similar performance when the pressure is smaller than $0.3kN$, and outperform the other methods. Our method causes slightly larger displacement compared to Polyhedral Voronoi and Crossfill when the pressure is between $0.3kN$ and $0.4kN$. However, our method has the smallest displacement among all methods under the other pressures. That means our infill structure has the best stiffness among these structures under most pressure conditions. Like some of the compared methods, the machine code of our infill structure can be generated in the construction of the infill structure. Therefore, we don't need to reconstruct the 3D infill structure and a slicing process, which simplifies the preprocessing of 3D printing. Since all objects can withstand the large pressures in the longitudinal direction, we do not conduct the physical tests in that direction.

It is worth noticing that the machine codes of (a), (b), (c), (e), and (f) are generated with the Cura engine [34], (d) is generated with the IceSL engine [38], and (g) is generated with the Slic3r engine [37]. We do not perform the physical simulations for Crossfill [6], Polyhedral Voronoi [5], Octet [30], Cubic [30], and 3Dhoney [37], as we can only obtain their Gcodes from the Cura engine, the IceSL engine, and the Slic3r engine, respectively. We observe from the practical printing tests that the layers of Crossfill [6] have a lot of sharp turns which largely affect the robustness and efficiency of the extrusion-based 3D printing technologies.

Some layers of 3Dhoney [37] contain overhanging paths which are difficult to print. In addition, all the above infill structures cannot give sufficient support for objects with overhanging inner surfaces. However, a subdivision strategy is proposed in this work to deal with such issue which will be described in Section 5.4.

Except for the above infill structures that have been integrated into 3D printing softwares, there are many other high performance infill structures. Among them, we mainly compare with two representative infill structures: the Honeycomb structure [3] and the Rhombic structure [7] under the same surface pressure $0.9N/mm^2$. Figure 15 and Figure 16 show the comparison results, where the models for the comparisons are provided by the corresponding authors or downloaded from the authors' homepage. To fairly compare with them, the fill ratios of the models generated with our method are the same as their models which are 40.5% in Figure 15 and 54.5% in Figure 16.

In Figure 15, we show the comparison results with [3]. The results of our method are generated with three different infill parameters which are: (a) $r = 11.5mm$ and $t = 1.5mm$, (b) $r = 9.8mm$ and $t = 1.0mm$, and (c) $r = 8.0mm$ and $t = 0.9mm$. The stress maps are generated by considering the loading conditions on the top of the models (shown with the blue arrows). The MVMSs of these models are 6.062MPa (a), 7.361MPa (b), 6.367MPa (c), and 8.867MPa (d), respectively. It can be proved that our infill structure has better stiffness than the Honeycomb structure, even under the different infill parameters. In addition, our structure is

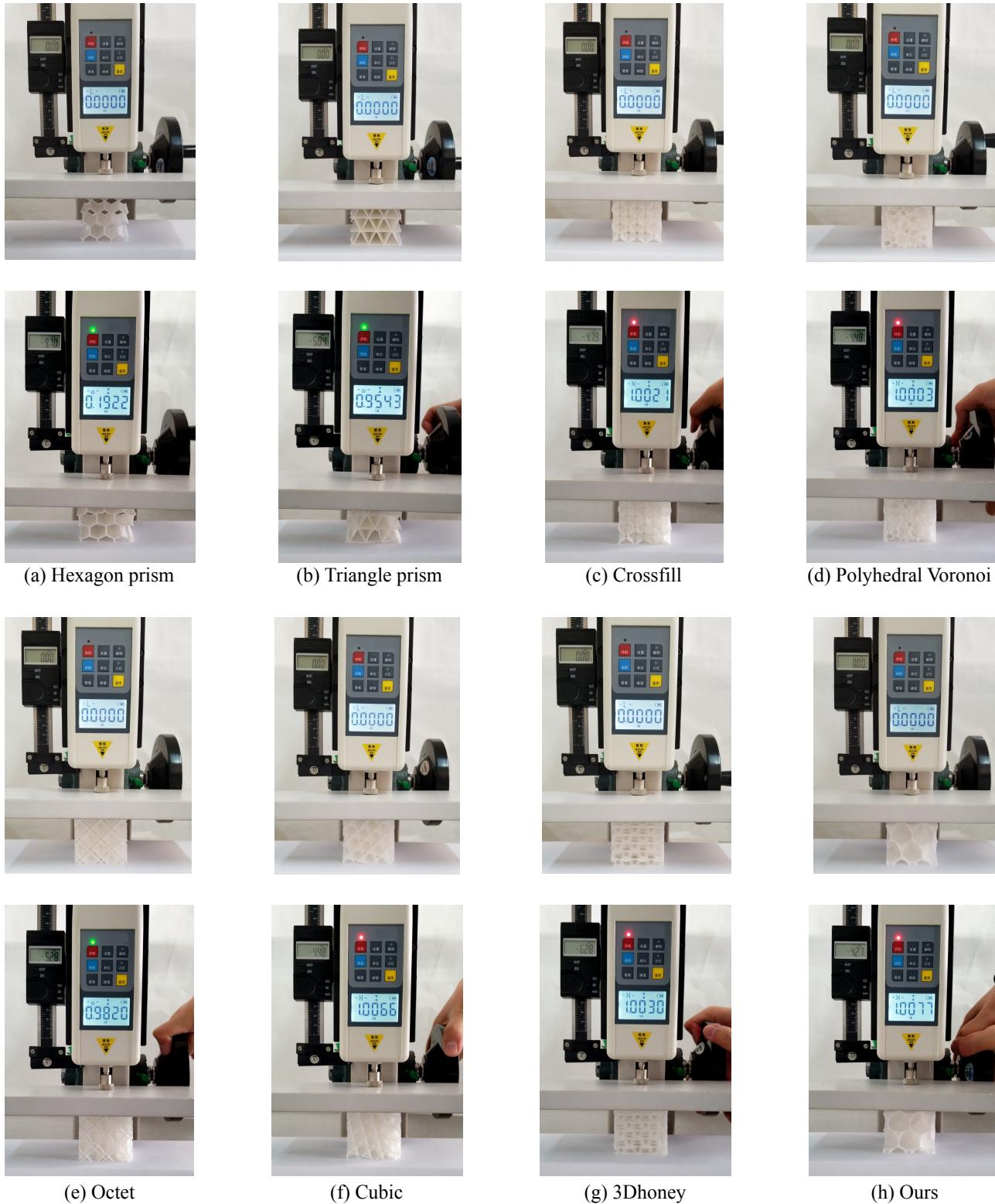


Fig. 13. Physical tests of the objects filled with the regular hexagonal prism [34] (a), the triangle prism [34] (b), Crossfill [6] (c), Polyhedral Voronoi [5] (d), Octet [30] (e), Cubic [30] (f), 3Dhoney [37] (g), and our infill structure (h). The first and the third rows show the initial status before testing. The second and the fourth rows show the status of the objects under different pressures: 0.1922kN (a), 0.9543kN (b), 1.0021kN (c), 1.0003kN (d), 0.9820kN (e), 1.0066kN (f), 1.0030kN (g), and 1.0077kN (h), respectively. The corresponding displacements are: 9.70mm (a), 5.04mm(b), 4.79mm (c), 4.48mm (d), 5.28mm (e), 4.48mm (f), 6.28mm (g), and 4.27mm (h), respectively.

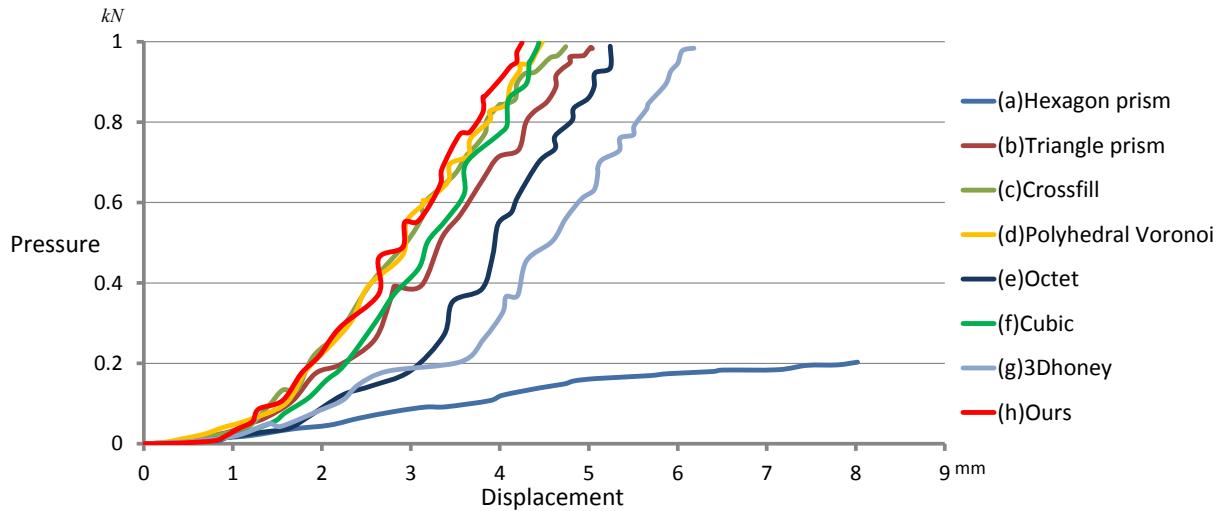


Fig. 14. The displacement-pressure curves of the physical tests for different methods shown in Figure 13.

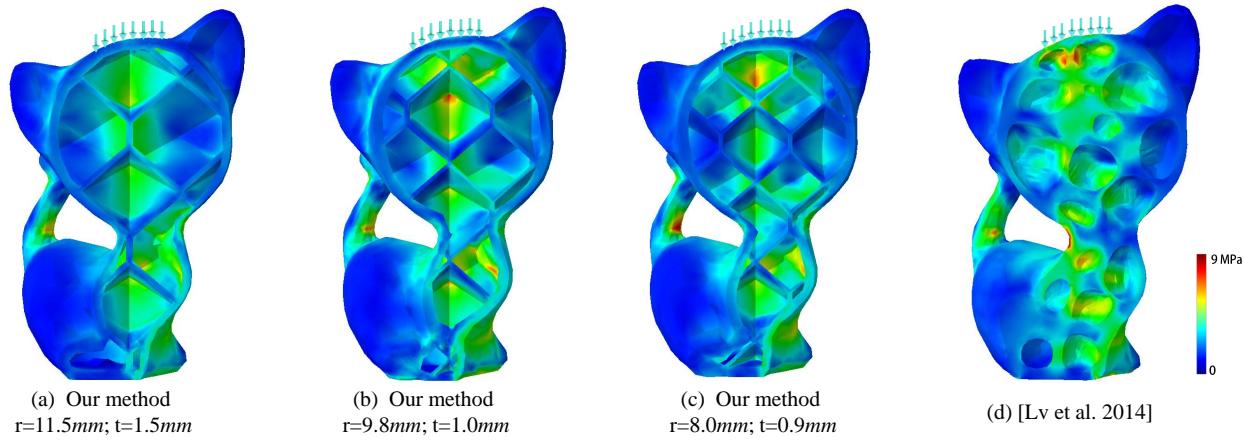


Fig. 15. Structural comparisons between [3] and our method with different parameters under the same fill ratio which is 40.5 %. MVMSs from left to right are: 6.062MPa (a), 7.361MPa (b), 6.367MPa(c) and 8.867MPa (d).

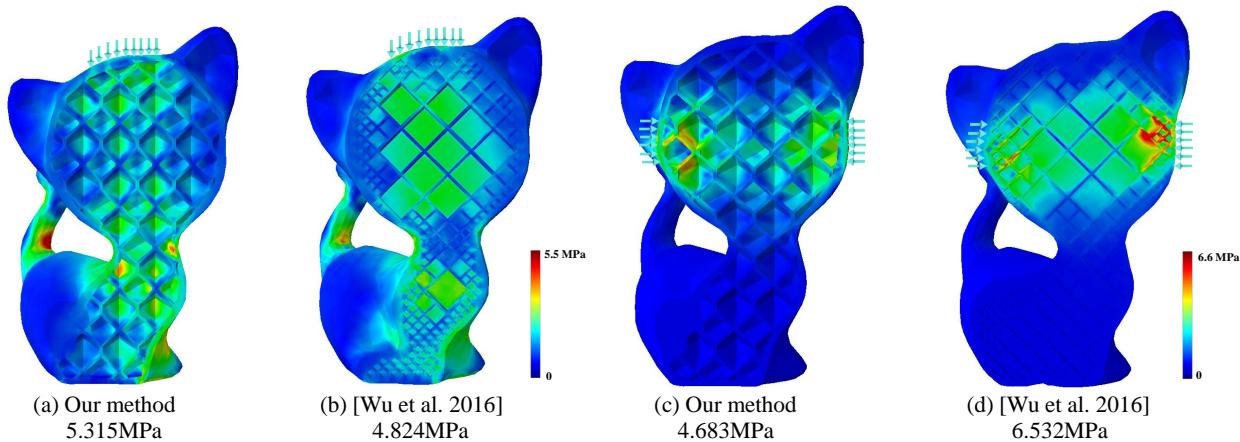


Fig. 16. Structural comparisons with [7] under the same fill ratio which is 54.5%. The structural performances are compared under two loading conditions: the longitudinal force ((a) and (b)) and the lateral force ((c) and (d)). MVMSs are listed at the bottom.

self-supporting while the Honeycomb structure is difficult to print with the extrusion-based 3D printers.

In Figure 16, we show the comparison results between our structure ((a) and (c)) and the Rhombic structure [7] ((b) and (d)). For (a) and (b), the forces are acted on the top of the models, and the MVMSs of both models are 5.315MPa (a) and 4.824MPa (b), respectively. MVMS of our structure is slightly larger than the Rhombic structure [7] as their structure is adaptively optimized under this loading condition. In (c) and (d), we show the comparison result under another loading condition. In this case, the MVMSs are 4.683MPa (c) and 6.532MPa (d). As we can see, the difference of MVMS between (a) and (c) of our infill structure is smaller than that of the Rhombic structure [7], which proves that our infill structure has balanced structural performance under different loading conditions. From the above comparisons we can conclude, our infill structure is self-supporting and easy to print, has balanced structural performance, and does not need an extra slicing process to generate the machine code, which reduces the preprocessing time of 3D printing.

5.3 Printed objects

In Figure 17, we show four different models filled with our infill structure. The first row shows the rendering results and the second row shows the printed objects. For these models, the infill structures are generated with different parameters r and t which are listed at the bottom. The models are cut to clearly see the internal infill structures.

5.4 Discussions

Anisotropy Through the above comparisons and analysis we can see, 3D models filled with our infill structure have comparable structural performance under different loading conditions. However, when force is applied along the printing direction and perpendicular to the printing direction, the layer-upon-layer 3D printing technologies will result in very different structural performances. This issue may be mitigated with the multi-axis 3D printing technologies, such as [39].

Subdivision As mentioned in Section (1), our infill structure is self-supporting, but it cannot give sufficient support when printing a model with overhanging inner surfaces if r is too large, for instance the roof of a cube. In this section, we propose a subdivision strategy to solve this issue. Our main motivation is that 3D printers can print an impending line with a certain length if two ends of this line are supported. The pipeline of the proposed strategy is shown in Figure 18.

Starting from a layer which is composed of equilateral triangles (a), all nodes on this layer move along the edges connected with them. After all nodes moved a distance s , a new cross-section (b) is generated, where the nodes on the original pattern (a) are fixed and the purple edges are newly inserted. When all nodes reach the middle of their corresponding edges, the whole pattern (a) is subdivided once (c). The process from (a) to (c) is called as a subdivision cycle. This cycle is iteratively performed until the maximal in-plane distance of the unit cell, which corresponds to the final equilateral triangle pattern, is smaller than the maximal printable impending length ε . Here the in-plane distance

means the distance between two nodes on the unit cell which belong to the same cross-section.

For a unit cell, the nodes on the hexagonal cross-section (the blue cross-section in Figure 5) have the maximal in-plane distance. That is the orange line shown in Figure 19. As we can see, if the edge length of the hexagon is r , the edge length of the corresponding equilateral triangle is $\sqrt{3}r$ and the maximal in-plane distance is $2r$. After n subdivisions (n should be an integer), the edge length of the equilateral triangle is $\sqrt{3}r/2^n$. Then, the maximal in-plane distance of the unit cell constructed with this equilateral triangle pattern is $r/2^{n-1}$. To ensure the support-free printing for the overhanging inner surfaces, the following inequality should be satisfied:

$$\frac{r}{2^{n-1}} \leq \varepsilon. \quad (5)$$

That is

$$n \geq \log_2\left(\frac{2r}{\varepsilon}\right). \quad (6)$$

As n is an integer, we can select it as

$$n = \lceil \log_2\left(\frac{2r}{\varepsilon}\right) \rceil, \quad (7)$$

where $\lceil x \rceil$ denotes the smallest integer that is larger than or equal to x .

Since the moving step size is s , the number of layers required for a subdivision cycle is calculated as:

$$N = \lceil \frac{\sqrt{3}r}{2s} \rceil. \quad (8)$$

Therefore, the total number of layers required for the subdivision is $n \times N$. To perform the subdivision, we first detect all overhanging inner surfaces and denote the node on the overhanging surfaces which has the minimal z coordinate as q . That is the lowest node on the overhanging inner surfaces. The layer containing q is denoted as L_q . Below L_q , we search for a layer L_s which is filled with the equilateral triangle pattern, and the number of layers between L_q and L_s should be equal to or larger than $n \times N$. Then, the layers between L_q and L_s are subdivided with the above strategy. And the layers above L_s are constructed, starting from the final equilateral triangle pattern, with the same way described in Section 4.1. If the number of layers below L_q is smaller than $n \times N$, we start constructing our infill structure with $r = \varepsilon/2$ at the first layer, which means the maximal in-plane distance of the constructed unit cell is ε . That is the overhanging inner surfaces can be well supported during the printing.

The sphere model shown in Figure 20 (a) has overhanging inner surfaces (the top and side surfaces). We subdivide it according to the above strategy to ensure the support-free printing of the overhanging areas. Three cross-sections in different stages are shown from left to right, that is before subdivision (a), in subdivision (b), and after subdivision (c). More cross-sections of this model can be seen in the accompanying video.

Construction periodicity In subdivision, we need to exactly start from a layer which is filled with the equilateral triangle pattern. Therefore, we require the construction process is periodic. That is the distance between p_i and $p_i^{J_2}$ shown in Figure 4 should be an integer multiple of

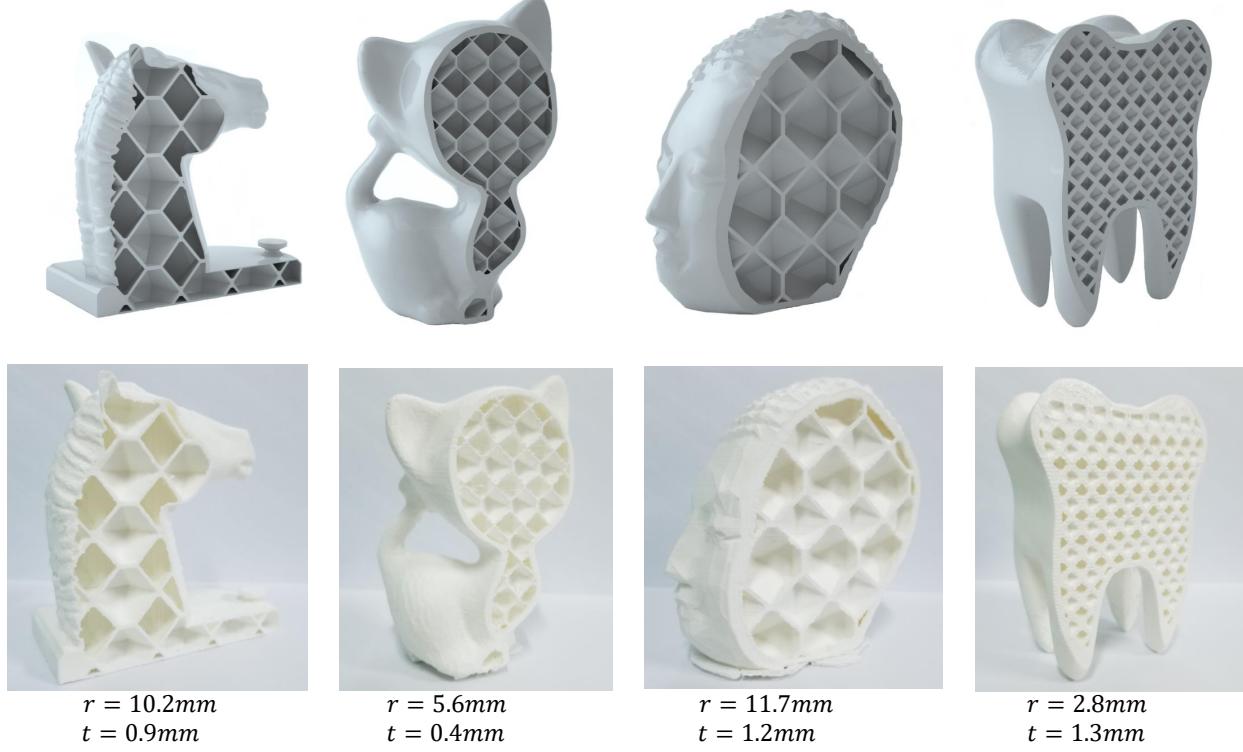


Fig. 17. A gallery of several printed objects filled with our infill structure. The objects are cut to show the interior infill structures. The edge length r and the edge thickness t for them are listed at the bottom.

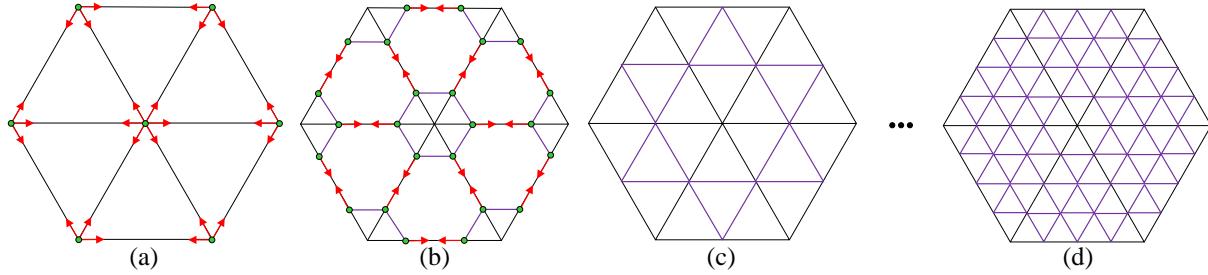


Fig. 18. The subdivision of a pattern filled with equilateral triangles. All nodes move along the edges associated with them to the middle of the corresponding edges (a). A new cross-section is generated after all nodes move a step size s (b). The whole pattern is uniformly subdivided once when all moving nodes meet at the middle of their corresponding edges (c). More subdivisions can be obtained by performing (a)-(c).(d) shows the result with two subdivisions.

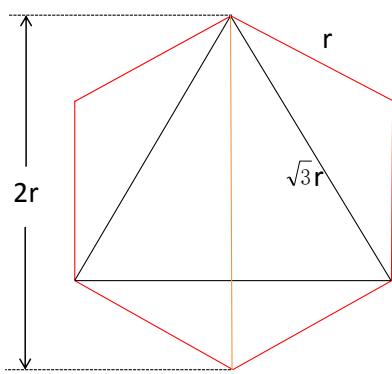


Fig. 19. The illustration of a hexagon and its maximal in-plane distance.

the moving step size s . As described in Section 4.3, s is related to the slicing thickness h . Therefore, we are able to adjust s and h to achieve the periodicity of the construction process while keeping the self-supporting property of the constructed infill structures.

Slicing time In Table 2, we list the slicing time of a cube model filled with different methods. The height of the cube model is 150mm, and the slicing thickness is 0.1mm. The fill ratio of this model is 50%. The models filled with different infill structures are sliced on a PC whose configuration is described in the first paragraph of Section 5. The slicing time of Rhombic [18], Honeycomb [3], Crossfill [6], Octet [30], and Cubic [30] are recorded from the Cura engine [34]. The slicing time of Polyhedral Voronoi [5] is recorded from the IceSL engine [38]. The slicing time of 3Dhoney [37] is recorded from the Slic3r engine [37]. For our method, the slicing time is the time for the generation of the Gcode

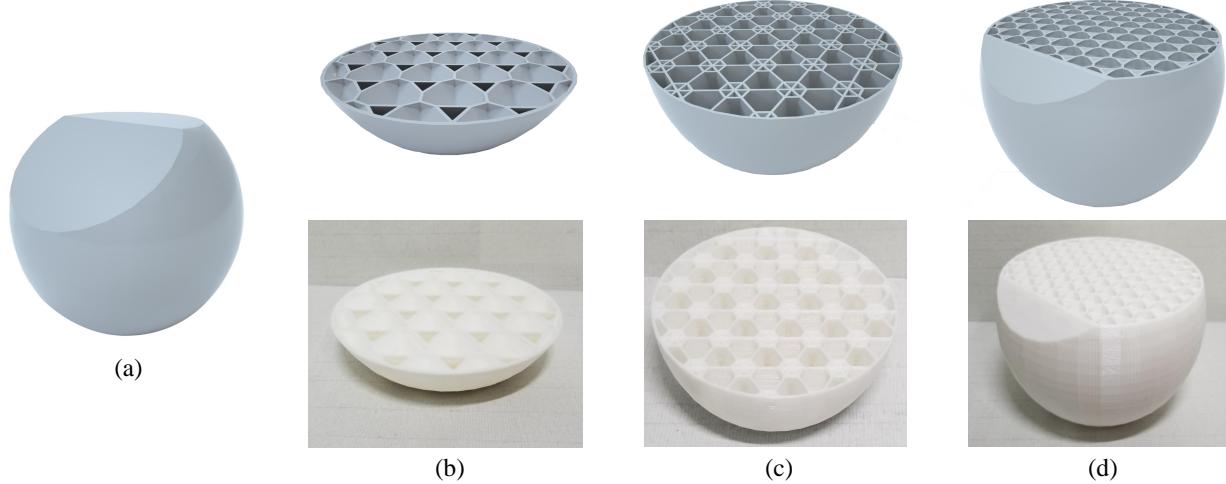


Fig. 20. A sphere model with an overhanging roof and an overhanging side surface (a) is subdivided with the proposed strategy. Three cross-sections in different stages are shown from left to right: (b) before subdivision, (c) in subdivision, and (d) after subdivision.

described in Section 4.1. From Table 2 we can see, Octet [30], Cubic [30], and our method have very similar slicing time. However, our method is the fastest among them. Here we should point out, the slicing time of different methods does not have much difference for the simple models and it is negligible compared to the printing time.

6 CONCLUSIONS AND FUTURE WORK

In this work, we propose a simple and novel light-weight support-free infill structure based on the layer construction to reduce the amount of printing material while maintaining a good structural stiffness for 3D models. The layers of our infill structure continuously and periodically transform between triangles and hexagons. The infill structures with different SSA can be generated by adjusting s to achieve the self-supporting properties for various 3D printing technologies. The machine code for 3D printing is automatically generated in the construction of the infill structure, which simplifies the preprocessing of 3D printing. A simple subdivision strategy is proposed to achieve the support-free printing of the model with overhanging inner surfaces. Physical simulations and comparisons demonstrate that our infill structure has comparable structural performance under different loading conditions and outperforms the existing infill structures in terms of structural stiffness. Furthermore, the relationship between the structural stiffness and the parameters of the infill structure is investigated which will be helpful for non-professional users.

For complex infill structures, the material leakage is a big issue for the extrusion-based 3D printing technologies. If each layer could be printed continuously, this issue would be well handled. In the next step, we will try to generate the continuous printing paths for our infill structure. A simple subdivision strategy is proposed to achieve the support-free printing of the overhanging inner surfaces. In each subdivision cycle, we need to subdivide the whole layer, as shown in Figure 18 (c). To make our method smarter, a local subdivision strategy according to the overhanging areas and physical analysis is required, which will be our future work. Since our method introduces a lot of voids in

3D models, it is only suitable for the extrusion-based 3D printing technologies, such as FDM. In the next step, we will try to propose a more general infill structure which is applicable for more types of 3D printing technologies. As the machine code of our infill structure is generated in the construction process, it will be easy to integrate our method into the commercial 3D printing software.

ACKNOWLEDGMENT

This work is partially supported by the Fundamental Research Funds for the Central Universities (DUT18RC(4)064), the Fundamental Research Funds for the Universities of Henan Province No.NSFRF180401, the Key Scientific Research Projects of Colleges and Universities of Henan Province No. 21A520017, the Natural Science Foundation of Henan Polytechnic University No. B2017-37, the Scientific and technological projects of Henan Province (182102210310, 182102210086), the Natural Science Foundation of China (U1811463, 61976040, 61762074, 61762064), and Jiangxi Science Fund for Distinguished Young Scholars (20192BCBL23001).

REFERENCES

- [1] J. Plocher and A. Panesar, "Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures," *Materials & Design*, p. 108164, 2019.
- [2] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu, "Cost-effective printing of 3d objects with skin-frame structures," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 177, 2013.
- [3] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, and B. Chen, "Build-to-last: strength to weight 3d printed objects," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 97, 2014.
- [4] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, and W. Wang, "Medial axis tree—an internal supporting structure for 3d printing," *Computer Aided Geometric Design*, vol. 35, pp. 149–162, 2015.
- [5] J. Martínez, S. Hornus, H. Song, and S. Lefebvre, "Polyhedral voronoi diagrams for additive manufacturing," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–15, 2018.
- [6] T. Kuipers, J. Wu, and C. C. Wang, "Crossfill: foam structures with graded density for continuous material extrusion," *Computer-Aided Design*, vol. 114, pp. 37–50, 2019.

TABLE 2
The statistics of the slicing time for the infill structures generated with different methods.

Methods	Rhombic [18]	Honeycomb [3]	Crossfill [6]	Polyhedral Voronoi [5]	Octet [30]	Cubic [30]	3Dhoney [37]	Ours
Timing (s)	26.64	19.66	36.86	248.89	6.96	6.24	47.91	5.69

[7] J. Wu, C. C. Wang, X. Zhang, and R. Westermann, "Self-supporting rhombic infill structures for additive manufacturing," *Computer-Aided Design*, vol. 80, pp. 32–42, 2016.
[8] W. Wang, S. Qian, L. Lin, B. Li, B. Yin, L. Liu, and X. Liu, "Support-free frame structures," *Computers & Graphics*, vol. 66, pp. 154–161, 2017.
[9] M. Lee, Q. Fang, Y. Cho, J. Ryu, L. Liu, and D.-S. Kim, "Support-free hollowing for 3d printing via voronoi diagram of ellipses," *Computer-Aided Design*, vol. 101, pp. 23–36, 2018.
[10] E. Vouga, M. Höbinger, J. Wallner, and H. Pottmann, "Design of self-supporting surfaces," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 87:1–87:11, Jul. 2012.
[11] Y. Liu, H. Pan, J. Snyder, W. Wang, and B. Guo, "Computing self-supporting surfaces by regular triangulation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 92:1–92:10, Jul. 2013.
[12] M. Deuss, D. Panozzo, E. Whiting, Y. Liu, P. Block, O. Sorkine-Hornung, and M. Pauly, "Assembling self-supporting structures," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 214:1–214:10, Nov. 2014.
[13] M. Livesu, S. Ellero, J. Martinez, S. Lefebvre, and M. Attene, "From 3d models to 3d prints: an overview of the processing pipeline," in *Computer Graphics Forum*, vol. 36, no. 2. Wiley Online Library, 2017, pp. 537–564.
[14] L. Liu, A. Shamir, C. C. Wang, and E. Whiting, "3d printing oriented design: geometry and optimization." in *SIGGRAPH ASIA Courses*, 2014, pp. 1–1.
[15] L. Liu, W. Xu, W. Wang, Z. Yang, and X. Liu, "Survey on geometric computing in 3d printing," *Chinese Journal of Computers*, vol. 37, no. 10, pp. 1–28, 2015.
[16] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, "The status, challenges, and future of additive manufacturing in engineering," *Computer-Aided Design*, vol. 69, pp. 65–89, 2015.
[17] A. M. e Sá, V. M. Mello, K. R. Echavarria, and D. Covill, "Adaptive voids," *The Visual Computer*, vol. 31, no. 6–8, pp. 799–808, 2015.
[18] J. Wu, N. Aage, R. Westermann, and O. Sigmund, "Infill optimization for additive manufacturing—approaching bone-like porous structures," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 2, pp. 1127–1140, 2017.
[19] T. Reiner and S. Lefebvre, "Interactive modeling of support-free shapes for fabrication," in *Proceedings of Eurographics 2016 - Short Papers*, 2016.
[20] W. Wang, Y.-J. Liu, J. Wu, S. Tian, C. C. Wang, L. Liu, and X. Liu, "Support-free hollowing," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 10, pp. 2787–2798, 2017.
[21] Y. Yang, S. Chai, and X.-M. Fu, "Computing interior support-free structure via hollow-to-fill construction," *Computers & Graphics*, vol. 70, pp. 148–156, 2018.
[22] Y. Xie and X. Chen, "Support-free interior carving for 3d printing," *Visual Informatics*, vol. 1, no. 1, pp. 9–15, 2017.
[23] A. Telea and A. Jalba, "Voxel-based assessment of printability of 3d shapes," in *International symposium on mathematical morphology and its applications to signal and image processing*. Springer, 2011, pp. 393–404.
[24] N. Umetani and R. Schmidt, "Cross-sectional structural analysis for 3d printing optimization," in *SIGGRAPH Asia Technical Briefs*. Citeseer, 2013, pp. 5–1.
[25] Q. Zhou, J. Panetta, and D. Zorin, "Worst-case structural analysis," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 137–1, 2013.
[26] O. Stava, J. Vanek, B. Benes, N. Carr, and R. Měch, "Stress relief: improving structural strength of 3d printable objects," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 48, 2012.
[27] Y. Xie, W. Xu, Y. Yang, X. Guo, and K. Zhou, "Agile structural analysis for fabrication-aware shape editing," *Computer Aided Geometric Design*, vol. 35, pp. 163–179, 2015.
[28] W.-P. Xu, W. Li, and L.-G. Liu, "Skeleton-sectional structural analysis for 3d printing," *Journal of Computer Science and Technology*, vol. 31, no. 3, pp. 439–449, 2016.

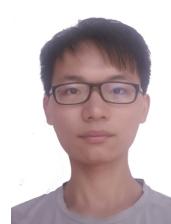
- [29] W. Wang, B. Li, S. Qian, Y.-J. Liu, C. C. Wang, L. Liu, B. Yin, and X. Liu, "Cross section-based hollowing and structural enhancement," *The Visual Computer*, vol. 33, no. 6–8, pp. 949–960, 2017.
[30] Cubic, "<https://all3dp.com/2/cura-infill-patterns-all-you-need-to-know/>," 2020.
[31] Slic3r, "<https://github.com/slic3r/slic3r/issues/1646>," 2020.
[32] C. C. Wang and Y. Chen, "Thickening freeform surfaces for solid fabrication," *Rapid Prototyping Journal*, vol. 19, no. 6, pp. 395–406, 2013.
[33] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Trans. Graph.*, vol. 11, no. 3, pp. 228–258, 1992.
[34] Cura, "<https://github.com/ultimaker/cura>," 2020.
[35] Abaqus, "<https://www.3ds.com/de/produkte-und-services/simulia/produkte/abaqus/>," 2019.
[36] Youngs, "<https://www.britannica.com/science/youngs-modulus>," 2020.
[37] Slic3r, "<https://slic3r.org/>," 2020.
[38] IceSL, "<https://icesl.loria.fr/>," 2020.
[39] C. Dai, C. C. Wang, C. Wu, S. Lefebvre, G. Fang, and Y.-J. Liu, "Support-free volume printing by multi-axis motion," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.



Wenpeng Xu is currently an associate professor in Henan Polytechnic University, School of Computer Science and Technology. He received his master degree in 2004 from Zhejiang University, China, and the PhD degree in 2016 from University of Science and Technology of China respectively. His research interests are Geometric design and computing for manufacturing.



Yi Liu is currently a master student at School of Computer Science and Technology in Henan Polytechnic University. He received his B.S. degree in computer science and technology from Hubei University of Arts and Science in 2017. His research interests are Geometric design and computing for manufacturing.



Menglin Yu is currently a undergraduate student at School of Mechanical and Power Engineering in Henan Polytechnic University. His research interests are Geometric design and computing for manufacturing.



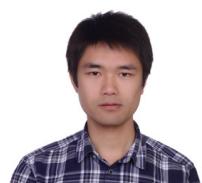
Dongxiao Wang is currently a master student at School of Computer Science and Technology in Henan Polytechnic University. She received her B.S degree in Internet of Things Engineering from Xinlian College of Henan Normal University in 2018. Her research interests are Geometric design and computing for manufacturing.



Shouming Hou is currently a professor in Henan Polytechnic University, School of Computer Science and Technology. He received his master degree in 2000 from Huazhong University of Science and Technology and the PhD degree in 2010 from Northeastern University, China, respectively. His research interests are Computer Graphics and CAD.



Bo Li is currently a professor in Nanchang Hangkong University, School of mathematical and information sciences. He received his B.S and PhD degree in 2002 and 2008 from the Dalian University of Technology. His research interests include the areas of Computer Graphics and Digital Geometric Processing.



Weiming Wang is currently a lecturer in Dalian University of Technology, Department of Mathematical Sciences. He received his B.S and PhD degrees in 2010 and 2016 from Dalian University of Technology. His research interests are Computer Graphics and 3D Printing.



Ligang Liu is a Professor at the School of Mathematical Sciences, University of Science and Technology of China. His research interests include digital geometric processing, computer graphics, and image processing. He serves as the associated editors for journals of IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Graphics and Applications, Computer Graphics Forum, Computer Aided Geometric Design, and The Visual Computer. He served as the conference co-chair of GMP 2017 and the program co-chairs of GMP 2018, CAD/Graphics 2017, CVM 2016, SGP 2015, and SPM 2014. His research works could be found at his research website: <http://staff.ustc.edu.cn/~lgliu>