Computers & Graphics 90 (2020) 108–115

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Section on SMI 2020

Ceramic 3D printed sweeping surfaces

Fanchao Zhong[a], Wenqiang Liu[b], Yu Zhou[a], Xin Yan[a], Yi Wan[b], Lin Lu[a,*]

[a] School of Computer Science and Technology, Shandong University, Qingdao, China
[b] Key Laboratory of High-Efficiency and Clean Mechanical Manufacture, College of Mechanical Engineering, Shandong University, Jinan, China

ARTICLE INFO

Article history:
Received 3 May 2020
Accepted 5 May 2020
Available online 17 May 2020

Keywords:
Sweeping surface
Ceramic 3D printing

ABSTRACT

Ceramic 3D printing has gained rapid development in recent years. Especially, desktop ceramic 3D printers get more and more popular for designers, artists, and makers to work with clays. As a highly viscous pseudoplastic liquid, the printing material clay produces featured aesthetics, and also indicates several manufacturing constraints. There are barely customized modeling tools for ceramic printing. Targeting for ceramic printing, we present a user-interactive modeling and fabricating framework using sweeping lines as the design tool. We integrate the ceramic printing constraints like self-supporting and collision-free in the modeling stage. Then we propose a customized method for generating the printing file, with the guarantee that the extruder traces a single continuous path without starts and stops. To fulfill this goal, we compute an adaptive zig-zag path along the sweeping surface. During the fabrication, our continuous path avoids the potential collision between the printing head and the physical model. Moreover, we optimize the extrusion amount along with the extruder movement, such that overlap between neighboring layers is appropriate in terms of both model stability and cost-effectiveness. We demonstrate the fabrication process and physical results to validate our method.

1. Introduction

Various ceramics materials have versatile and excellent properties, like high mechanical strength and hardness, good thermal and chemical stability and viable thermal, optical, electrical, and magnetic performance [1]. Therefore they are used in a wide range of applications, including the chemical industry, electronics, aerospace, biomedical engineering, and decorations. With the rapid development of additive manufacturing, there are different ceramic 3D printing techniques like binder jetting, stereolithography, and material deposition etc. Among them, material deposition, also known as robocasting (RC), or direct ink writing (DIW), is the most common technique [2]. It can be regarded as a modification of fused deposition modeling (FDM) technique, but with extruders adapted to the semi-liquid pastes. This technique takes clay as the material and therefore is of the least expensive cost and fast manufacturing process. The fabrication process is also similar to the traditional "coiling" technique [3], which has been used to shape clay into vessels for many thousands of years.

There are many open-source projects for building desktop ceramic 3D printers specialized for clay as the material. Such consumer-level 3D printers get more and more popular with designers, artists, and makers who like to work with clays. However, there are barely modeling tools customized for ceramic for inexperienced designers. Most printed ceramic artworks are of simple-shape, like vases and potteries. Therefore, we aim at proposing a user-interactive modeling and fabrication framework for ceramic 3D printing that can provide enough design space for normal users. Considering the ceramic property, we focus on the surface rather than volume modeling in this attempt.

As a viscous and soft paste, clay induces special constraints for 3D printing. Specifically, clay maintains the same soft consistency during the entire printing process. As elaborated in [1], when moving the print head without extruding, the paste will not immediately be detached from the model and thus cause deformations and printing failure. Therefore, a single continuous print path without transfer moves for the entire model should be guaranteed.

We utilize the sweeping surfaces as the modeling tool and bridge the modeling and fabrication stages considering the fabrication constraints. That is, we directly translate the users' design intention into a legal printing file for extrusion-based ceramic 3D printers. We customize both the printing tool path and the extrusion amount such that the printed model is valid, and no collision or collapse would happen in the printing process.

The main contributions of our approach are as follows:

- We introduce a user-interactive sweeping surface modeling scheme that integrates the ceramic printing constraints. The

* Corresponding author.
E-mail address: llu@sdu.edu.cn (L. Lu).

https://doi.org/10.1016/j.cag.2020.05.007
0097-8493/

users input two profile curves and one trajectory curve as the design intention; our framework corrects the curves if they violate the self-supporting or non-self-intersection constraints and then directly translate the design into a valid G-code file.
- We propose a novel printing file generation algorithm for ceramic 3D printing, which produces a single continuous printing tool path without transfer moves. Considering the characteristics of clay, we present the adaptive zig-zag path generation method and control the extrusion amount along the continuous path, to guarantee the continuity and avoid collapse.

2. Related work

Lightweight Modeling for Fabrication Along with the rapid development of 3D printing techniques, as well as the printing materials, modeling tools are still in the developing stage. Design tools for inexperienced users with no expert knowledge or skills on 3D modeling are of high demand.

Much research effort has been devoted to lightening the 3D modeling process and integrating fabrication constraints in the modeling stage. Recent works consider different kinds of fabrication techniques and produce customized wire jewelries [4], lampshades [5], wire sculptures [6] etc. For comprehensive surveys of the literature, we refer to [7,8].

Sketching is an essential interactive modeling method, which has been facilitated by the large amount of research work. Bae et al. [9] proposed a sketching interface using digitized pen strokes to simulate the pen and paper curve modeling process. Tim et al. [10] defined brushes to realize an interactive sculpting approach that enables modeling of support-free objects. Recent methods integrate geometric principles or deep-learning to guide the modeling of freeform surfaces from sketches [11–13], or 2D silhouettes [14].

As a physical sketching tool, a 3D extruder pen is also utilized in personal fabrication [15]. Yue et al. [16] built a mixed reality system to guide the wire sculpturing using a 3D pen.

Interactive Sweeping Modeling As a useful curve sketching and modeling tool, sweeping is a standard command in CAD modeling systems like AutoCAD, SolidWorks [17], SketchUp [18], and SKETCH [19]. A sweeping surface is created by extending a profile curve along a specified path.

Fowler et al. [20] presented a comprehensive model of seashells determined by sweeping a generating curve along the helicospiral. Chen et al. [21] modelled the model based on the user-defined 2D profile and main axis of the 3D object in the input image. Li et al. [22] developed a sketch-based modeling interface, which takes a pair of user strokes as input and instantly generates a curved 3D surface by sweeping one stroke along the other. Deng et al. [23] proposed lofted shapes modeling from a single image and user input strokes. However, the above methods aim at geometric modeling and barely consider the fabrication constraints, like printability, self-support, etc.

Vinayak and Ramani [24] proposed an interactive modeling tool for potteries, using hand grasp and motion to express the user's intention. Such models can be fabricated for ceramic printing, but the shapes are too simple to be attractive. Different from the above methods, we aim at bridging the modeling and fabricating stages, providing the users with enough design freedom, and directly produce the printing file.

Printing File Computation A 3D printer uses a numerically controlled programming language made up of a series of commands, so-called G-Code. These commands tell the printer exactly what actions to perform - moving path, speed, extrusion amount, temperatures, etc. Efficient tool path is crucial for material extrusion-based additive manufacturing [25], which affects both the printing efficiency and effectiveness. Zhao et al. [26] proposed a region fill algorithm using connected Fermat spirals with global continuity, which has good performance for complex shapes or interior structures. They further applied the spirals to subtractive manufacturing [27]. Jin et al. [28] developed a path filling pattern to avoid the retraction during the pasted materials deposition process. Etienne et al. [29] introduced curved toolpaths for standard 3-axis printers to print curved objects, to reduce the staircase defects.

The printing path should be carefully modulated in robotic fabrication system. One of the critical issues in such a continuous extrusion printing scheme is to avoid the collisions during motion. Wu et al. [30] computed collision-free tool paths for printing wire mesh models. Huang et al. [31] further considered stability constraints jointly with the collision-free constraints. Dai et al. [32] fabricated solid models along variational directions on 5-DOF volume printing. They computed the feasible sequence of material deposition for the given model considering the path continuity, collision-free, and support-free constraints. Bhatt et al. [33] established a two-manipulator system including a 3-DOF build-platform and a 3-DOF extrusion tool to build free-form thin shell parts.

The extrusion amount can also be modulated to generate aesthetic expressions of the printed models [34,35].

Fabrication constraints imposed by printable clay is similar, but more "strict", i.e., the print head is extruding during the whole printing process. Hergel et al. [1] proposed a method for integrated tool path planning and support structure generation tailored to the extrusion-based ceramics printing. The authors introduced proxy geometry that safeguards the model and allows the deposition paths to navigate around the model while maintaining a close but safe distance. In our case, our fabricated model is a shell surface. Thus we customize the printing file by optimizing both the tool path and the extrusion amount to ensure the printability and enhance the stability. To the best of our knowledge, we are the first to customize the G-code file in terms of both toolpath and extrusion amount for ceramic printing.

3. Overview

Our framework includes two stages, user-interactive modeling, and printing file generation.

The users are asked to draw two profile curves C_s, C_t and one trajectory curve Ψ with the center o located at the origin on the 2D canvas (Fig. 1(a-b)). We assume the printing direction is z-axis, C_s and C_t starting with zero z-value. The trajectory Ψ is defined in the xoy-plane. The sweeping surface is determined by the revolution from C_s to C_t along with Ψ about z-axis, as shown in Fig. 2.

Once the three curves are settled by the user, we generate the sample points of the underlying sweeping surface and perform an automatic local refinement on the sample points, such that the model is support-free and self-intersection-free, i.e., ready for clay 3D printing. Then we generate the printing file with no need to reconstruct the surface mesh. We propose a novel toolpath generation algorithm specified for ceramic printing with clay, in which both the trajectories of the extrusion head and the extrusion amount are variables and carefully computed.

Our toolpath is tailored for the open sweeping surface with no fill-in, considering fabrication constraints for extrusion-based ceramic printing, as follows.

- The clay is continuously extruded, i.e., the deposition path is a strictly single continuous path with no stop.
- There is no collision between the extrusion head and printed model in the whole printing process.
- The model stands in printing with no extra support structures.
- The overlap between clay layers is sufficient and cost-effective.

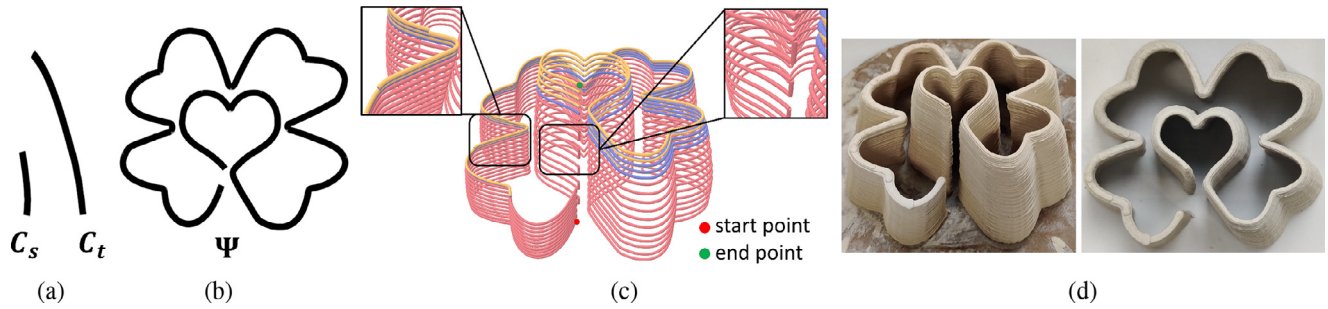

Fig. 1. The modeling and printing pipeline. Given two profile curves C_s, C_t (a) and the trajectory curve Ψ (b), we formulate the sweeping surface and translate it into the printing file with a non-stop continuous path and adaptive extrusion amount, targeting for the extrusion-based ceramic printing process (c), such that the model can be fabricated correctly (d). In the whole process, we do not need to reconstruct the surface model.

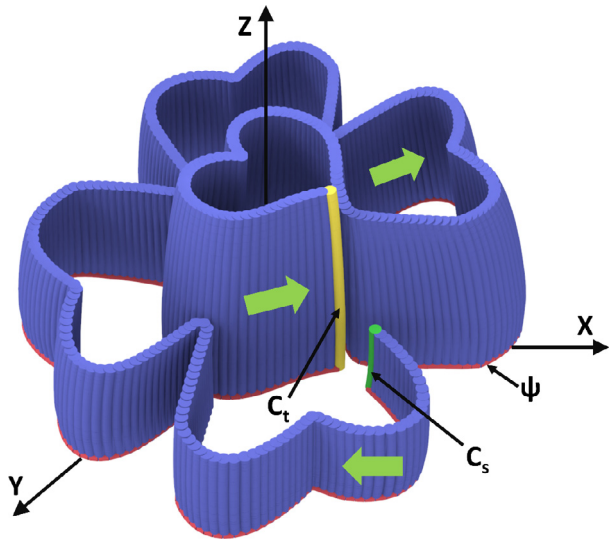

Fig. 2. The sweeping surface is controlled by two profile curves C_s (green) and C_t (yellow), the trajectory curve Ψ (red) around z-axis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The path generation is performed based on tracing the sample points along the sweeping surface in a zig-zag manner. To avoid the collision between the extruder and the printed model, we ensure that the height difference between the two corresponding points on C_s and C_t is smaller than the height of the extruder. To extend between the two profile curves with the height difference larger than the extruder, we compute the adaptive returning points along the path. Such that the adaptive zig-zag path is computed (Fig. 1(c)).

The extrusion amount is initially set as the minimal value, and then the amount is optimized according to the overlap between layers, such that no collapse would happen.

Finally, we generate the printing file of the sweeping surface for ceramic 3D printing and fabricate the physical object (Fig. 1(d)).

4. Sweeping surface modeling

We develop a lightweight modeling interface for common users that are not familiar with the CAD software rich of professional knowledge. The users draw two profile curves C_s, C_t and one trajectory curve Ψ using the clamped cubic B-spline curve on the 2D canvas (Fig. 2).

Moreover, we provide a control curve for users to edit the interpolation function to have more freedom on the surface shape.

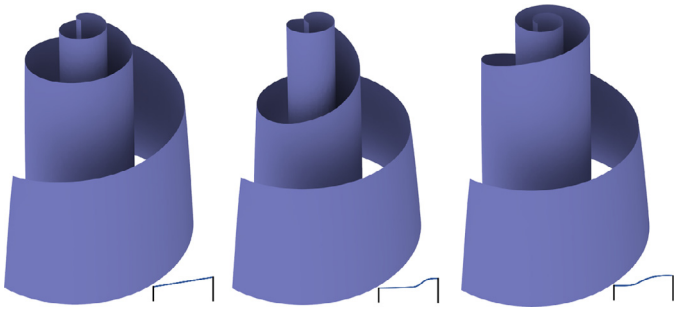

Fig. 3. The sweeping surfaces with different interpolation functions between profile curves, shown in the bottom right corner for each shape. All the other parameters are identical for the three shapes.

The interpolation function determines how C_s sweeps to C_t along Ψ, as demonstrated in Fig. 3. By default, the function is set as linear, as shown in the leftmost subfigure. Note that we only allow monotonic functions here, to avoid the possible collisions during the fabrication due to equipment limitations. We also allow the users to take a pre-defined curve as an input Ψ, as long as it is continuous and self-intersection free.

Regarding to the constraints for ceramic 3D printing, the sweeping surface should be self-supporting and without self-intersection. Therefore, in addition to ensuring the monotonicity of profile curves along the printing direction (z-axis) and no intersection among C_s, C_t and Ψ, we automatically tune the profile curves such that they are support free and there is no self-intersection for the whole sweeping surface. To perform the curve modulation, we generate sample points to represent the sweeping surface.

Sample points generation For two profile curves C_s and C_t, we denote the height by $Z(C_s)$ and $Z(C_t)$, respectively. Denote the minimum layer height by Δz_0. We equidistantly sample $\min(Z(C_s), Z(C_t))/\Delta z_0$ points on C_s and C_t, respectively, obtaining two sample sets V_s and V_t. We also equidistantly sample the trajectory curve Ψ into n points. Then we apply interpolation between V_s and V_t along Ψ following the user-specified interpolation function, producing $n-2$ transition curves represented by intermediate points on each transition curve. Points in V_s, V_t, and intermediate points of transition curves have a one-to-one correspondence in z-axis order.

Support-free correction The overhang angle threshold for clay is larger than that of thermoplastic materials in extrusion-based printing techniques. In our experiments, this threshold is 57° (see discussions in Section 5.3). We locally modify two profile curves and all transition curves if there exist points that violate the overhang threshold.

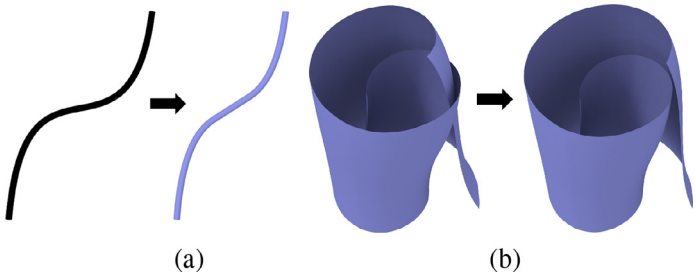

Fig. 4. User input correction. (a) The profile curve violating the support-free constraint is automatically corrected. (b) Self-intersection in the sweeping surface is detected and treated.

Specifically, for each profile curve or transition curve, we traverse the sample points along z-axis from $z = 0$. For each sample point, we compute its slope angle, i.e., the angle between the line connecting the point and its neighbor and z-axis. If the slope angle is larger than the overhang angle threshold, we compute the new position of the sample point that satisfies the threshold and drag it to the new position. Then we apply the same translation operation to the following points on the curve. The same detection and treatment are applied to all curves, starting from C_s to C_t along the transition curves. An example of the correction is shown in Fig. 4 (a).

Self-intersection detection and treatment Self-intersection may occur even though C_s, C_t and Ψ are of no self-intersections (Fig. 4 (b)). Therefore, the self-intersection detection for the whole surface is required.

We traverse any two points on Ψ, and if the straight line connecting them is close to the origin o, we apply further check for their transition curves. If the distance between two intermediate points on the transition curves is approximately zero, the sweeping surface is self-intersected.

We iteratively compress C_s and C_t along x-axis until the self-intersection is eliminated. See the example in Fig. 4 (b). We note that our treatment here is brute-force and there may be alternatives for eliminating the intersection with less modifications of the entire shape.

5. Printing file generation

5.1. Continuous fabrication constraints

We use direct ink writing (DIW) technique for the 3D ceramic printing. The characteristics of clay induce that we need a single continuous printing path for the entire model, in which the extrusion is continuous without starts and stops.

We take advantages of the clay material that the single path is stable enough, and only fabricate the sweeping surface with no fill-in structures. I.e., we tend to compute the continuous tool path along with the sweeping surface, like a curvy wall.

Since our sweeping surface is an open model, the conventional layer by layer slicing and path planning tools cannot guarantee the continuity of the path. Sometimes, errors will even be reported when slicing. Therefore, we need to customize the printing file generation algorithm for the designs.

G-code format G-code is a series of commands to control a 3D printer, telling the 3D printer exactly what actions to perform, such as where to move, what speed to use, what temperatures to set, and much more.

Here we show a sample G-code segment:

G0 F2400 X54.0445 Y163.005 Z1.16
G1 F2000 X57.9786 Y162.8205 Z1.3200 E3.2500
G1 X61.9337 Y162.5500 Z1.4800 E5.2073

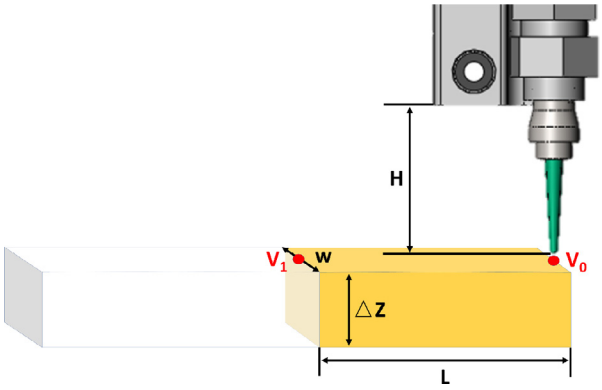

Fig. 5. Illustration for extrusion amount calculation.

G0 is a command for non-extrusion rapid linear movements. G1 command tells the printer to move in a straight line to the location that X, Y, or Z values specify. The E value corresponds to the extrusion amount of the material. The F value defines the movement speed of the printer, specified in units of mm/min.

In this work, we compute the path as well as the extrusion amount along each step.

Fabrication parameters To generate the G-code file, the printing parameters are taken into account, including the height of the extruder H, the minimum layer height Δz_0, the minimum line width $w_{\min}$, and the maximum line width $w_{\max}$. The line width represents the width of the printing path. Fig. 5 is favorable for users to understand the specific content of these parameters. These parameters are mainly determined by the inner diameter of the extruder. In our setting, $\Delta z_0 = 1.25mm$, $w_{\min} = 3mm$ and $w_{\max} = 6mm$.

In our algorithm, we compute the actual extrusion amount e of every point on the printing path, illustrated in Fig. 5. Here, V_0 and V_1 indicate two neighboring printing points on the same layer of the printing path, i.e., a moving step, and the step size is L. The average line width for V_0 and V_1 is denoted by w, and the average layer height is Δz. The final extrusion amount e on V_0 is approximated by the volume of the yellow cuboid. The E value in G-code is computed by dividing e by the rotating speed of extrusion screw, which determines the clay extrusion flow.

The printing file generation algorithm is given in Algorithm 1.

Algorithm 1 Printing file generation.

Input: The surface parameters C_s, C_t, Ψ and fabrication parameters $H, \Delta z_0, w_{\min}, w_{\max}$;
Output: The G-code file.
1: Initialize the G-code file;
2: Adaptive zig-zag Path Generation $(C_s, C_t, \Psi, H, \Delta z_0)$;
3: Adaptive Extrusion Amount Computation $(P, w_{\min}, w_{\max})$;
4: Wrap up the printing file;
5: **return** The printing file.

Given the surface parameters and fabrication parameters, the algorithm generates the G-code file for ceramic printing. There are two key components: first settle the sample points to represent the sweeping surface and control the extruder path, and then determine the extrusion amount for each move step.

5.2. Adaptive zig-zag path generation

We first generate the sample points based on the design curves to represent the sweeping surface and then compute the continuous toolpath. Due to the possible collision during the fabrication process, we propose a two-stage path strategy.

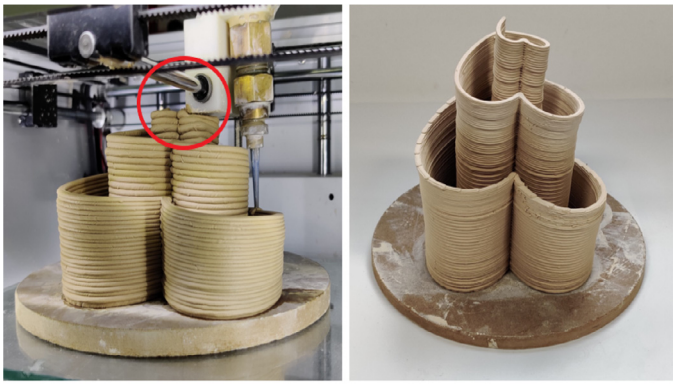

Fig. 6. Collisions avoidance during the extrusion head movement. (a) Collisions occur in the manufacturing; (b) The model is successfully printed with our printing path considering collision avoidance.

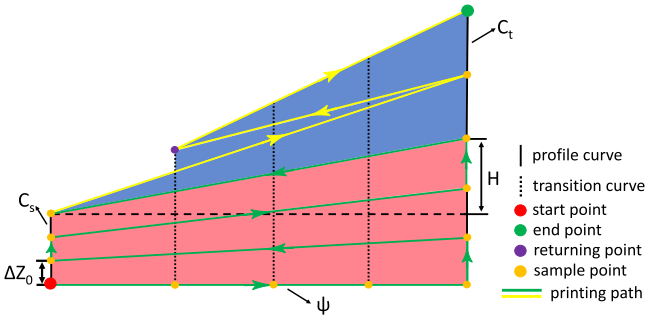

Fig. 7. The printing path generated by our algorithm. Green path indicates the lower part; yellow path indicates the upper part. Both sets of paths form the single continuous zig-zag tool path..

Collisions in the fabrication Since the extruder follows a continuous toolpath along the sweeping surface (instead of layer by layer), collisions between the extrusion head and the printed model may happen during the fabrication process. As shown in Fig. 6(a), such accident occurs when the height difference of C_s and C_t in z-axis is larger than H. To avoid this collision, we propose a two-stage path strategy that keeps the height difference of any two points in the same printing layer lower than H.

The basic idea is to divide the path into two parts and generate the adaptive zig-zag path, respectively. For the convenience of illustration, we assume C_s is shorter than C_t. An illustration is given in Fig. 7, in which the trajectory is a straight line.

We divide C_t into two parts at z-value $Z(C_s)+H$, referring to the lower and the upper parts (red and blue) in Fig. 7. For C_s and the lower part of C_t, we equidistantly sample them using the parameter Δz_0 into the same number of points, respectively, as described in *sample points generation* above. We connect the sample points in the same layer which has one-to-one correspondence generating the zig-zag tool path along the trajectory, shown as the green printing path in Fig. 7. During printing, the height position of the extruder varies continuously in the same printing layer with the height changes of two corresponding endpoints.

For the upper part, we follow the steps below. We calculate z-axis values of *returning points* by adding up the sampling distance of V_t to $Z(C_s)$. For every calculated z-axis value, we look up the highest point of every transition curve, which has the nearest distance with it as the returning point, such as the purple point in Fig. 7. We start from the end of the last path, generating the zig-zag tool path according to the un-printed sample points of V_t and returning points. The yellow printing path in Fig. 7 is generated in this way. Refer to Algorithm 2 for the details.

Fig. 8. Adaptive extrusion amount computation. (a) Insufficient extrusion amount causes instability and model collapse; (b) Adaptive extrusion amount produces stable results.

Algorithm 2 Adaptive Zig-zag Path Generation.

Input: The surface parameters C_s, C_t, Ψ and the nozzle height H, the minimum layer height Δz_0;
Output: The G-code file.

```
Generate the sample points based on C_s, C_t, Ψ;
if |Z(C_t) − Z(C_s)| < H then
    Build the zig-zag path P;
else
    Divide the higher curve into two parts at z-value min(Z(C_s), Z(C_t)) + H;
    Generate the re-sample points based on the lower part of the higher profile, the lowerprofile and Ψ;
    Build the zig-zag path P_l for the lower part;
    Compute returning points on transition curves;
    Build the zig-zag path P_u between returning points and remaining sample points of the higher profile curve;
    Combine two path segments together P ← P_l + P_u;
end if
Update the G-code file.
```

5.3. Adaptive extrusion amount computation

Because the extrusion amount of clay for ceramic printing is generally larger than other materials, the extrusion amount for each move step is also a critical issue. Insufficient extrusion amount would cause less support for adjacent layers and lead to model collapse, as shown in Fig. 8(a). On the other hand, too much extrusion amount causes waste of materials, uncontrolled surface quality and more fabrication time and thus uncertainty. Therefore, we propose the adaptive extrusion amount computation method.

As illustrated in Fig. 9, the horizontal distance of two neighboring printing points of two neighboring layers is represented by s, the overlap width of two neighboring layers is represented by Δa, and w is the line width. The angle θ is between the printing direction and the tangent vector of the point.

Since the clay will solidify in a short time, the subsequent printing layer is not easy to collapse due to superposition. Therefore, we consider the previous layers are safe from collapse and only optimize on the line width of two neighboring layers. We remark that this assumption only holds in a local range of layers. In our experiments, this strategy works as the models are relatively small (mostly height below 100mm) due to the limited building

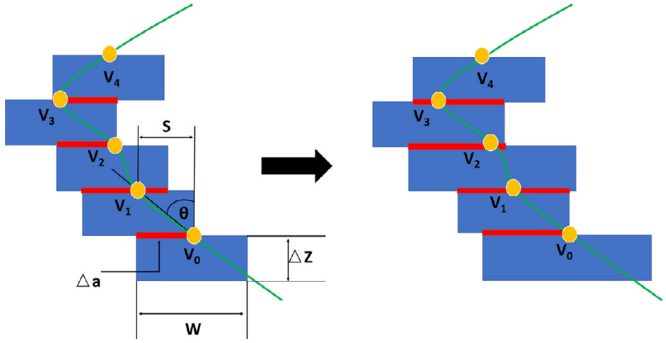

Fig. 9. Illustration on the adaptive extrusion amount computation.

volume size. However, if the height of the models goes large, stress analysis for the whole model would be necessary.

Our strategy is to ensure more than half of the line width of the current layer or its gravity center lying on its previous neighboring layer. For the printing path we computed, we assume that the z-axis motion of the extruder is unchanged, and only adjust the line width. The line width is proportional to the clay extrusion flow. Here, we set the overlap proportion threshold as 2/3, which means the overlap part of the current layer and its previous neighboring layer accounts for at least 2/3 of the width of the current layer. This ratio is an empirical value based on experiments.

In our framework, the initial width of the printing path for the whole model is always set to be $w_{\min}$. Traversing every printing point from the bottom layer to the top layer, we keep the line width of the current point and previously visited point of the neighboring layer equal, while expanding or reducing their width simultaneously with the same scale until the proportion of overlap reaches 2/3. It is often necessary to adjust the width of every printing point of each layer twice. If the calculated line width is less than $w_{\min}$, it is reset to $w_{\min}$.

After traversal, the overlap part of every layer and its previous neighboring layer accounts for greater than 1/2 of the width of the current layer.

When we achieve the printing path and its line width, we can compute the extrusion amount e for every point during printing. The detailed algorithm is described in Algorithm 3. Fig. 8 (b) shows an example after the extrusion amount adjustment.

Algorithm 3 Adaptive Extrusion Amount Computation.

Input: Printing path P, line width parameters $w_{\min}$ and $w_{\max}$;
Output: The G-code file.

```
i ← 1;
for each V_i ∈ P do
    s_i ← projected distance of |V_i − V_{i−1}| on xoy-plane;
    if 3s_i < w_min then
        w_i ← w_min;
    else
        w_i ← 3s_i;
    end if
    w_{i−1} ← w_i;
end for
Compute the extrusion amount e for each printing point;
Update the G-code file.
```

According to fabrication tolerances and the line width adjustment method we designed, the threshold of the overhang angle can be calculated. When s is greater than $1mm$, the line width is equal to s three times. The line width ranges from 3 mm to 6 mm, so s ranges from 0 mm to 2 mm. Refer to Fig. 9, we have $tan\theta = s/\Delta z$. The minimum value of Δz is 1.25 mm, so we get the overhang angle θ ranging from 0° to 57°.

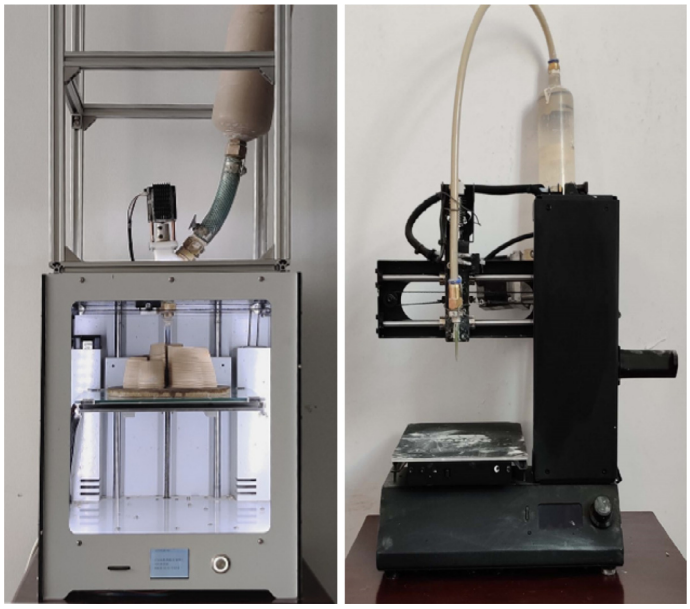

Fig. 10. Our ceramic 3D printers. CERAMBOT Plus (left), CERAMBOT Mini (right) [36]. The building volume is 180*190*200mm, 120*135*100mm (width*depth*height), respectively.

Table 1
Computational statistics of six user designed and fabricated models. For each model created by users, we list its physical size (Size), number of total sample points (#SamplePts.) and the fabrication time (Time) in minutes. The design curves and fabricated results are in Fig. 11, in which the models are labelled as t1 to t6 in a top-down order.

Model	Size (mm)	#SamplePts.	Time (min.)
t1	132.3 × 139.0 × 102.5	43724	33
t2	124.3 × 120.3 × 60.4	17420	13
t3	148.3 × 151.7 × 113.2	39215	25
t4	64.5 × 75.6 × 47.3	6842	5
t5	115.4 × 81.2 × 14.6	5736	4
t6	111.6 × 79.4 × 53.6	10944	7

6. Results and discussions

We implemented our algorithm in C++ with a 2.5 GHZ CPU. To evaluate our modeling interface, we invited ten undergraduate students to perform the surface design. It takes 2–10 minutes for users to finish designing the profile curves and the trajectory curve, depending on the complexity of the trajectory curves. Our path generation algorithm produces the G-code file in less than five seconds.

We picked six samples from the user created surfaces and fabricated them with our two XYZ-type desktop 3D printers (Fig. 10). The printed results and related statistics are shown in Fig. 11 and Table 1. The total number of sampling points is determined by both vertical sampling on the profile curves and horizontal sampling on the trajectory curve. As mentioned above, we set the minimal sampling rate on the lower profile curve as Δz_0=1.25 mm in the experiments. The horizontal sampling rate on the trajectory is empirically set as $1.1mm$ in the paper. The height of the extruder we use in the experiment is $H = 30mm$.

The results show that our modeling tool can realize various styles of surfaces and the quality of physical models printed by ceramic printers is generally satisfactory. We label the models in Fig. 11 as t1–t6 in a top-down order.

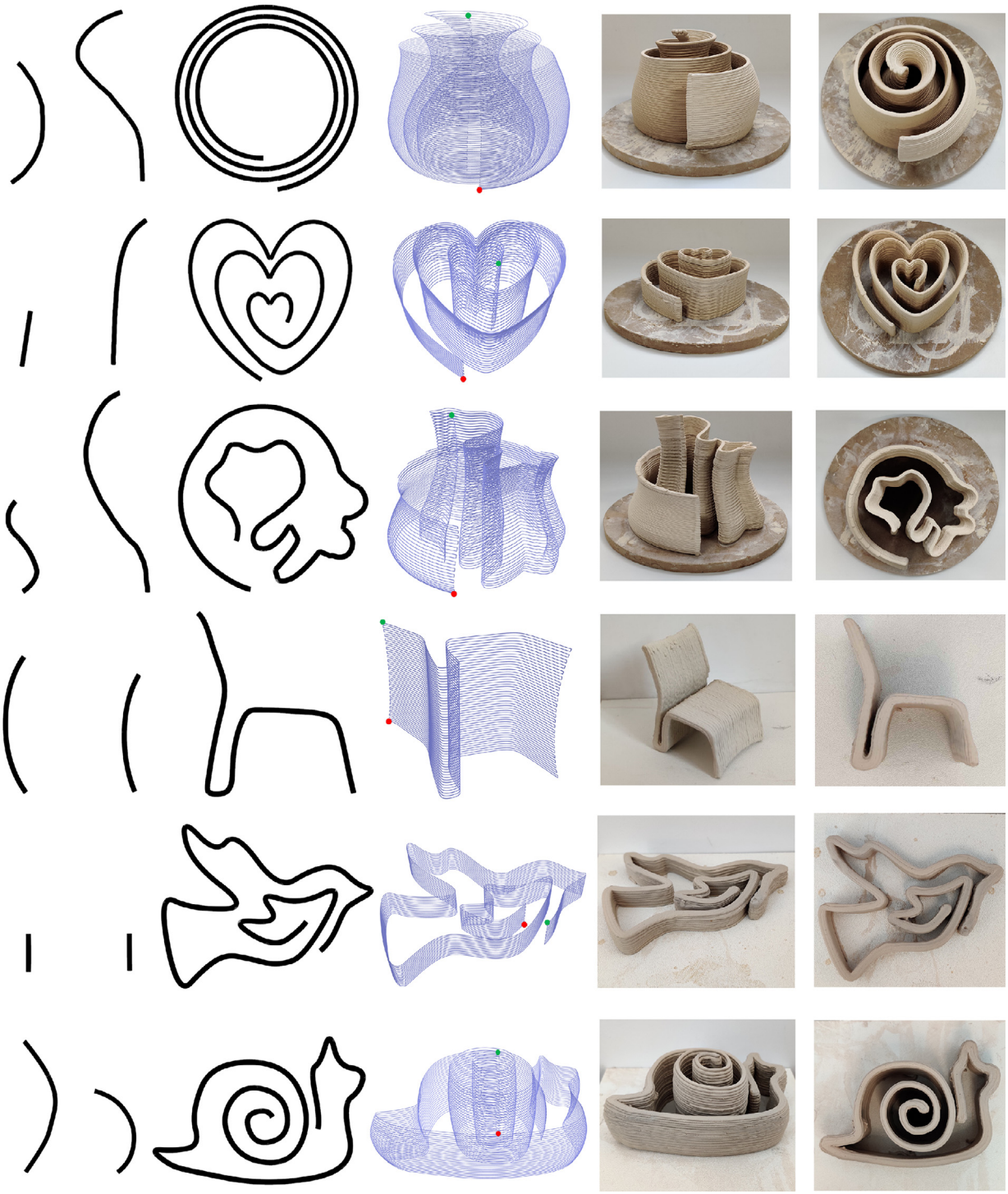

Fig. 11. Sweeping surface modeling and corresponding ceramic 3D printing results for six models (rows). Left-to-right columns: two profile curves, the trajectory curve, the printing path generated by our algorithm (red and green dots represent the start point and end point, respectively), and corresponding physical model of ceramic 3D printing in side and front views, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The trajectory curve in t1 is a user imported predefined curve, the others are drawn by the cubic B-spline on the canvas. From t2 to t6, the number of control points for the trajectory curve is 73, 71, 20, 31, 30, respectively.

Our modeling framework has no constraints on the height of the two profile curves. The height of C_s is less than that of C_t for t1, t2 and t3. The height of C_s is greater than that of C_t for t4 and t6 and the height of C_s and C_t is very close for the model t5.

For models t2 and t3, the height difference of the two profile curves is greater than H. Therefore, the two-stage path strategy is applied in their tool path.

The snail model (t6) benefits from the non-linear interpolation function between the profile curves, as the shell is thicker than the head.

To finish the whole ceramic printing process, we dry the printed model for twelve hours, and fire it in a 1245° electric pot-

tery kiln. The final result is stable and can be used in daily life. The size of the whole model will be reduced by nearly 10–20%.

We note that since our models are all open surfaces, we can still see the clay accumulation on the profile curves where the extruder turnarounds. This might can be further alleviated by tuning the speed of the extruder.

We also point out that our modeling and printing in this work mainly target for personalized design for decorative stuff. Hence, we have not considered the mechanical constraints like the global stiffness in the modeling framework. Standard structure analysis and enhancement techniques can apply if needed.

7. Conclusions and future work

In this paper, we propose a novel user-interactive modeling tool tailored for ceramic 3D printing. Our method generates a single continuous, self-supporting, collision-free deposition path for ceramic printing without reconstructing the mesh surface. The results show that our path generation approach improves on the range of models that can be printed in clay, and the printing process is reliable.

We believe that customized ceramic printing still has broad prospects and would like to work with professional designers to consider more intuitive design tools. Developing design tools for ceramics in a virtual environment is also an interesting direction.

We admit that the print quality for ceramics cannot match that of 3D printing with thermoplastic materials. However, we would consider transferring this quality "inaccuracy" into customized surface textures, by further adapting the printing parameters. Such surface textures can involve more functionalities for the personalized model, like grip and tactilities, etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Fanchao Zhong: Methodology, Software. **Wenqiang Liu:** Validation, Visualization. **Yu Zhou:** Writing - original draft. **Xin Yan:** Validation. **Yi Wan:** Resources. **Lin Lu:** Conceptualization, Writing - review & editing, Supervision.

Acknowledgments

We thank all the anonymous reviewers for their valuable suggestions. This work is supported by grants from NSFC (61972232), State Key Laboratory of Virtual Reality Technology and Systems (VRLAB2019A01), Key basic research project of Natural Science Foundation in Shandong province (ZR2018ZB0106) and the Key Science and Technology Innovation Project of Shandong Province (2019JZZY010112).

References

[1] Hergel J, Hinz K, Lefebvre S, Thomaszewski B. Extrusion-based ceramics printing with strictly-continuous deposition. ACM Trans Graph) 2019;38(6):1–11. doi:10.1145/3355089.3356509.
[2] Chen Z, Li Z, Li J, Liu C, Lao C, Fu Y, et al. 3d printing of ceramics: a review. J Eur Ceram Soc 2019;39(4):661–87. doi:10.1016/j.jeurceramsoc.2018.11.013.
[3] Coiling W.. https://en.wikipedia.org/wiki/Coiling_(pottery); 2020.
[4] Iarussi E, Li W, Bousseau A. WrapIt: computer-assisted crafting of wire wrapped jewelry. ACM Trans Graph 2015;34(6):1–8. doi:10.1145/2816795.2818118.
[5] Zhao H, Lu L, Wei Y, Lischinski D, Sharf A, Cohen-Or D, et al. Printed perforated lampshades for continuous projective images. ACM Trans Graph 2016;35(5) 154:1–154:11. doi:10.1145/2907049.
[6] Lira W, Fu C-W, Zhang H. Fabricable eulerian wires for 3d shape abstraction. ACM Trans Graph 2019;37(6):1–13. doi:10.1145/3272127.3275049.
[7] Livesu M, Ellero S, Martínez J, Lefebvre S, Attene M. From 3d models to 3d prints: an overview of the processing pipeline. Comput Graph Forum 2017;36(2):537–64. doi:10.1111/cgf.13147.
[8] Attene M, Livesu M, Lefebvre S, Funkhouser T, Rusinkiewicz S, Ellero S, et al. Design, representations, and processing for additive manufacturing. Synth Lect Vis Comput 2018;10(2):1–146. doi:10.2200/s00847ed1v01y201804vcp031.
[9] Bae S-H, Balakrishnan R, Singh K. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In: Proceedings of the 21st annual ACM symposium on User interface software and technology – UIST'08. ACM Press; 2008.
[10] Tim R, Sylvain L. Interactive modeling of support-free shapes for fabrication. In: Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers. Eurographics Association; 2016. p. 25–8. doi:10.2312/egsh.20161006.
[11] Li C, Pan H, Liu Y, Tong X, Sheffer A, Wang W. BendSketch: modeling freeform surfaces through 2d sketching. ACM Trans Graph 2017;36(4):1–14. doi:10.1145/3072959.3073632.
[12] Li C, Pan H, Liu Y, Tong X, Sheffer A, Wang W. Robust flow-guided neural prediction for sketch-based freeform surface modeling. ACM Trans Graph 2018;37(6):1–12. doi:10.1145/3272127.3275051.
[13] Smirnov D, Bessmeltsev M, Solomon J. Deep sketch-based modeling of man–made shapes. CoRR 2019. http://arxiv.org/abs/1906.12337
[14] Rivers A, Durand F, Igarashi T. 3d modeling with silhouettes. ACM Trans Graph 2010;29(4):1. doi:10.1145/1778765.1778846.
[15] Takahashi H, Kim J. 3d pen 3d printer: exploring the role of humans and fabrication machines in creative making. In: Proceedings of the 2019 CHI conference on human factors in computing systems – CHI '19. ACM Press; 2019.
[16] Yue Y-T, Zhang X, Yang Y, Ren G, Choi Y-K, Wang W. WireDraw: 3d wire sculpturing guided with mixed reality. In: Proceedings of the 2017 CHI conference on human factors in computing systems – CHI '17. ACM Press; 2017.
[17] Systèmes D.. Solidworks 2019. 2019.
[18] Trimble. Sketchup. 2020.
[19] Zeleznik RC, Herndon KP, Hughes JF. SKETCH: an interface for sketching 3d scenes. ACM SIGGRAPH 2006 courses on – SIGGRAPH '06. ACM Press; 2006. doi:10.1145/1185657.1185770.
[20] Fowler DR, Meinhardt H, Prusinkiewicz P. Modeling seashells. ACM SIGGRAPH Comput Graph 1992;26(2):379–87. doi:10.1145/142920.134096.
[21] Chen T, Zhu Z, Shamir A, Hu S-M, Cohen-Or D. 3-weep: extracting editable objects from a single photo. ACM Trans Graph 2013;32(6):1–10. doi:10.1145/2508363.2508378.
[22] Li Y, Luo X, Zheng Y, Xu P, Fu H. SweepCanvas: sketch-based 3d prototyping on an RGB-D image. In: Proceedings of the 30th annual ACM symposium on user interface software and technology – UIST '17. ACM Press; 2017.
[23] Deng C, Huang J, Yang Y-L. Interactive modeling of lofted shapes from a single image. Comput Vis Media 2019. doi:10.1007/s41095-019-0153-0.
[24] Vinayak, Ramani K. Extracting hand grasp and motion for intent expression in mid-air shape deformation: a concrete and iterative exploration through a virtual pottery application. Comput Graph 2016;55:143–56. doi:10.1016/j.cag.2015.10.012.
[25] Jin Y-A, He Y, Fu J-Z, Gan W-F, Lin Z-W. Optimization of tool-path generation for material extrusion-based additive manufacturing technology. Addit Manuf 2014;1–4:32–47. doi:10.1016/j.addma.2014.08.004.
[26] Zhao H, Chen B, Gu F, Huang Q-X, Garcia J, Chen Y, et al. Connected fermat spirals for layered fabrication. ACM Trans Graph 2016;35(4):1–10. doi:10.1145/2897824.2925958.
[27] Zhao H, Zhang H, Xin S, Deng Y, Tu C, Wang W, et al. DSCarver: decompose-and-spiral-carve for subtractive manufacturing. ACM Trans Graph 2018;37(4):1–14. doi:10.1145/3197517.3201338.
[28] Jin Y, He Y, Fu G, Zhang A, Du J. A non-retraction path planning approach for extrusion-based additive manufacturing. Robot Comput Integr Manuf 2017;48:132–44. doi:10.1016/j.rcim.2017.03.008.
[29] Etienne J, Ray N, Panozzo D, Hornus S, Wang CCL, Martínez J, et al. CurviSlicer: slightly curved slicing for 3-axis printers. ACM Trans Graph 2019;38(4):1–11. doi:10.1145/3306346.3323022.
[30] Wu R, Peng H, Guimbretière F, Marschner S. Printing arbitrary meshes with a 5dof wireframe printer. ACM Trans Graph 2016;35(4):1–9. doi:10.1145/2897824.2925966.
[31] Huang Y, Zhang J, Hu X, Song G, Liu Z, Yu L, et al. Framefab: robotic fabrication of frame shapes. ACM Trans Graph 2016;35(6):1–11. doi:10.1145/2980179.2982401.
[32] Dai C, Wang CCL, Wu C, Lefebvre S, Fang G, Liu Y-J. Support-free volume printing by multi-axis motion. ACM Trans Graph 2018;37(4):1–14. doi:10.1145/3197517.3201342.
[33] Bhatt PM, Malhan RK, Rajendran P, Gupta SK. Building free-form thin shell parts using supportless extrusion-based additive manufacturing. Addit Manuf 2020;32:101003. doi:10.1016/j.addma.2019.101003.
[34] Takahashi H, Miyashita H. Expressive fused deposition modeling by controlling extruder height and extrusion amount. In: Proceedings of the 2017 CHI conference on human factors in computing systems – CHI '17. ACM Press; 2017.
[35] Yaman U, Dolen M, Hoffmann C. Generation of patterned indentations for additive manufacturing technologies. IISE Trans 2018;51(2):209–17. doi:10.1080/24725854.2018.1491076.
[36] Cerambot. Cerambot. https://www.cerambot.com/; 2020.