Formation of Patterned Indentations for Additive Manufacturing Applications

Ulas Yaman Melik Dolen
Department of Mechanical Engineering
Middle East Technical University
Ankara, Turkey
{uyaman, dolen}@metu.edu.tr

Christoph Hoffmann
Department of Computer Science
Purdue University
West Lafayette, IN, USA
cmh@purdue.edu

Abstract—**This paper proposes a novel approach to generate patterned indentations for different additive manufacturing methodologies. Surface textures have many practices in different fields and require special manufacturing considerations. Beside conventional manufacturing processes, additive processes have also been utilized in the last decade to obtain textured surfaces. The current design and fabrication pipeline of additive manufacturing operations have many disadvantages in terms of surface patterns. For instance, the size of the design files grow considerably when there are detailed indentations on the surfaces of the artifacts. The presented method overcomes most of these disadvantages while fabricating the objects with surface patterns. It employs Boolean operations on binary images of the slices on the hardware of the 3D printers to obtain various types of textures.**

Keywords—surface textures; indentation; 3D printing; additive manufacturing; FDM; DLP

I. Introduction

Surface textures are the geometric features made in 3D on the surfaces of the artifacts [1]. They are used for a multitude of purposes in industry and academia. For instance, in mechanical engineering domain, they can be utilized:

- to increase surface areas for better heat transfer,
- to attain favorable tribological properties [1],
- to increase contact area (like knurling) for better handling (or grasp),
- to write relevant information (name, date, number, etc.) associated with the artifact,
- to increase the aesthetic appeal of the products,
- or specifically to imitate the surface properties of bones in orthopedic implants [2].

Additive Manufacturing (AM) methods are more convenient for creating surface textures than conventional manufacturing techniques. Although the conventional design and fabrication pipeline of AM methods has several drawbacks [3-4], it is widely used in related industries. Hence, the object of this study is to address the issue of surface texturing in the AM pipeline.

After the STL (STereoLithography) file is generated at the AM design stage, it is a difficult endeavor to add surface textures or to modify interior structures of the artifact at hand. The user must oftentimes revisit the design stage of the pipeline and add features that are available in the Computer Aided Design (CAD) software and follow the rest of the pipeline to fabricate the desired part. This is obviously a time-consuming task. Furthermore, to the best of the authors' knowledge, the commercial CAD packages do not offer any utilities/tools to create the patterned indentations discussed in this paper. Should there be any problems (no matter how small) associated with the outcome, the whole process must be reinitiated. Furthermore, it is a well-known fact that after embedding a large number of small features to the design, the size of the STL files grows considerably. Hence, slicing these huge models in the Computer Aided Manufacturing (CAM) software of the AM pipeline adds to the overall fabrication time. More than that, this brute force approach risks (physical) memory overflow and complicates memory allocation.

In this study, a different approach is proposed to overcome the above-mentioned problems. Instead of generating NC-code ("G code") files from a sliced CAD model that explicitly represents surface textures, the proposed method stores the texture information inside the printer instead. Specifically, the texture is represented by a sequence of binary bitmap images that will be manipulated by fast and cheap image processing algorithms. Since the Boolean operations on binary images are not computationally costly, the desired texture properties can be easily incorporated to the images on the fly, inside the printer, using cheap commodity PCs with limited resources. The details of the proposed method are discussed in the following sections of the paper and demonstrated using a Fused Deposition Modeling (FDM) and a Digital Light Processing (DLP) types of 3D printers.

The rest of the paper is organized as follows. After a brief introduction of texture printing in AM applications, the relevant literature is reviewed in the next section. The proposed approach is presented in the third section, which is followed by the test cases using different AM methods. Finally, the paper is concluded with key remarks and discussions of the possible future works.

II. Background

The practical importance of surface textures has motivated secondary manufacturing methods to create these textures on the surfaces of the artifacts such as painting, coating, engraving, etc.

This work is supported in part by the Scientific and Technological Research Council of Turkey under the project contract 116E882.

In terms of surface textures, the major advantage of AM methods over conventional manufacturing operations is that patterns could be potentially obtained without secondary operations [5]. Hence, AM is preferred for the manufacturing of complex geometric patterns [6]. For the colored patterns, AM is mostly used to obtain physical models first. Then different approaches are employed to color the artifacts. There are two recent examples of this approach; texture mapping with hydrographics [7] and computational thermoforming [8].

Since there are many disadvantages of the current design and fabrication pipeline of AM systems, researchers are working on new paradigms to overcome these drawbacks and extend the capabilities of AM machinery [9]. In one of these approaches, Vidimce et al. [10] proposed a programmable pipeline for multi-material fabrication. Their pipeline is not only used for different material distributions, but also for generating various patterns on the surfaces of the objects. Once a specific pattern is designed in their programming language, it can be used on different objects. This approach improves the flexibility of the AM machinery without using additional hardware.

Researchers have found different ways to obtain surface textures utilizing FDM processes over the last few years. Extrusion stage and the form (filament) of the material used in the FDM method make it easier to achieve various types of patterns. In one of the related studies, Takahashi and Miyashita [11] have drawn patterns by adjusting the two main printing parameters: **i)** clearance between the extruder tip and the corresponding layer; **ii)** the amount of extruded material. For embossing letters on the vertical walls, they just adjusted the extrusion rate. This approach resembles the one presented by Kanada [12]. The major difference in Kanada's study [12] is that the patterns are generated via spiral/helical printing technique in which the whole path is seamless. A totally different method was adapted by Herpt [13] to obtain surface patterns on FDM printers. He gives excitations to the printer from the ground and there is no other modification on the FDM printer. Due to the periodic oscillations, the extruder yields textured surfaces.

Although stringing is a major problem in FDM printers for regular prints, several researchers have used this property as a tool to fabricate textured surfaces. For instance, Laput et al. [14] created hair like structures by modifying the parameters of the extrusion process. Likewise, Ou et al. [15] utilized the same approach to produce dense hair patterns to attach different parts together.

On the other hand, it is much easier to create patterns (in color) on surfaces if the FDM printer has dual extruders or a single extruder fed by multiple colored filaments. Reiner et al. [16] made good use of a dual extruder FDM printer to obtain interleaved color patterns. They were able to fabricate a specific image on curved surfaces with this unique approach. Multi-jet 3D printers are superior to FDM printers in terms of color printing as different compositions of basic colors are impelled by jets. Brunton et al. [17] expanded the capabilities of multi-jet printers by employing an error diffusion approach. They did generate layers inside the artifacts to get finer color details.

Although there are many *ad hoc* approaches to generate surface textures on FDM printers, there is no specific solution for DLP type of 3D printers. In this study, we have proposed a method especially designed for DLP printers. However, it can also be used for different AM technologies.

III. Proposed Method

The paper presents easy-to-implement methods to generate patterned indentations on a wide variety of 3D objects. The methods discussed here are primarily based on morphological operations on binary images and could be adapted to most 3D printing technologies. Details follow:

A. Basic Morphological Operations on Binary Images

In image processing, morphological operations (such as dilation and erosion) are frequently employed to enlarge or shrink certain patterns in binary images. For that purpose, a mask is continuously applied throughout the contour of the selected pattern. Depending on the formation of the mask and the subsequent logical operations performed on the image, the desired outcome is obtained [18].

To be specific, let $\mathbf{I} \in \mathbb{B}^{N \times M}$ be a binary image (N by M pixels) representing a particular cross-section of a 3D-printed object where $\mathbb{B} \in \{0, 1\}$ denotes the Boolean set. The *erosion* operation applied at a particular location/point $\mathbf{p} = (\mathrm{x}, \mathrm{y}) \in \mathbb{N}^2$ in the image sets the bits to zero inside the circular region centered at p with radius $r \in \mathbb{N}$:

$$\mathbf{I}_{\boldsymbol{\alpha}(x),\boldsymbol{\beta}(y)} \coloneqq \mathbf{I}_{\boldsymbol{\alpha}(x),\boldsymbol{\beta}(y)} \wedge \mathbf{T} \tag{1}$$

where $\mathbf{I}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is the sub-matrix (sub-image) of **I** indexed by the sets $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (for columns and rows respectively) while the elements of the tool/mask matrix $\mathbf{T} = [t_{ij}] \in \mathbb{B}^{\mathbf{n} \times \mathbf{n}}$ $(n \equiv 2r + 1)$ become

$$t_{ij} = \begin{cases} 0, & \left\lfloor (i - r - 1)^2 + (j - r - 1)^2 + \frac{1}{2} \right\rfloor \leq r^2 \\ 1, & else \end{cases} \tag{2}$$

where $\lfloor \cdot \rfloor$ denotes *floor* (i.e. round to the lowest integer) function. Note that in accordance with **T**, the index sets in (1) are defined as

$$\boldsymbol{\alpha}(x) = \mathbb{N}_{\geq x-r}^{\leq x+r+1}; \boldsymbol{\beta}(x) = \mathbb{N}_{\geq y-r}^{\leq y+r+1} \tag{3}$$

Similarly, the dilation operation can be expressed as follows:

$$\mathbf{I}_{\boldsymbol{\alpha}(x),\boldsymbol{\beta}(y)} \coloneqq \mathbf{I}_{\boldsymbol{\alpha}(x),\boldsymbol{\beta}(y)} \vee (\neg \mathbf{T}) \tag{4}$$

Hence, one can apply the Boolean operations defined in (1) and (4) along the contours of the pattern(s) in **I** to obtain the desired transformation. To that end, boundary tracing algorithms [19] can be employed to find the pixel locations at the outer periphery of the pattern(s) contained in the image:

$$\mathbf{P} = \left\{ \mathbf{p_i} = (x_i, y_i) \in \mathbb{N}^2 \middle| i \in \mathbb{N}_{n_p} \right\} \tag{5}$$

where n_p denotes the number of points on the boundary. When the operation in (4) [or (5)] is conducted for all the points in (5), the pattern(s) in **I** is simply eroded (or dilated) by *r* pixels.

B. Patterned Indentations on Objects

Using morphological operations discussed in the previous sub-section, one can conveniently create a number of patterned indentations over the surface of the 3D-printed objects without the intervention of the CAD software provided that the cross-sectional images are readily available. Let **S** be a set of images representing the successive cross-sections of a 3D printed object:

$$\mathbf{S} = \{\mathbf{I}_k \in \mathbb{B}^{N\times M} | k \in \mathbb{N}_{\leq K}\} \tag{6}$$

where K is the number of images in this set which are set by L_z pixels apart (in vertical/z direction). For this purpose, a texture function Φ(x,y,z) ($\Phi{:}\mathbb{N}^3 \rightarrow \mathbb{B}$) needs to be defined to indicate the locations of the features in the image space (or the domain). If Φ(x,y,z) returns 1 (true), the pixel at (x,y) in $\mathbf{I_k}$ ($k = \lfloor z/L_z \rfloor$) pertains to this feature (i.e. indentation). Notice that this approach harbors the notion that the 3D object is to be "sculptured" out of a fictitious textured (i.e. porous) material. However, well-known texture mapping techniques in the literature [20] could be utilized for the applications where a 2D image needs to be mapped onto the surface of the printed object. As a simple texture function, 3D checkers array (tilted around principal axes) can be considered:

$$\Phi(x,y,z) = \begin{cases} 1, & u = 2\left\lfloor \frac{u}{2} \right\rfloor \\ 0, & else \end{cases} \tag{7a}$$

$$\begin{gathered} u \triangleq \lfloor x'/N_x \rfloor + \lfloor y'/N_y \rfloor + \lfloor z'/N_z \rfloor \\ \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c\theta_y c\theta_z & -c\theta_y s\theta_z & s\theta_y \\ s\theta_x s\theta_y c\theta_z + c\theta_x s\theta_z & -s\theta_x s\theta_y s\theta_z + c\theta_x c\theta_z & -s\theta_x c\theta_y \\ -c\theta_x s\theta_y c\theta_z + s\theta_x s\theta_z & c\theta_x s\theta_y s\theta_z + s\theta_x c\theta_z & c\theta_x c\theta_y \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{gathered} \tag{7b}$$

where N_x, N_y, N_z refer to the dimensions of the cells (in pixels) along the fundamental directions while $s\theta_* = \sin(\theta_*)$; $c\theta_* = \cos(\theta_*)$; (* is a placeholder for x, y, z). Note that in (7b), θ_x, θ_y, θ_z refer to the rotations around the major axes. Other checkers arrays could be formed using different coordinate systems. In cylindrical coordinate system, a "polar" checkers array could be easily produced. Hence, (7b) takes the following form:

$$\begin{gathered} u \triangleq \lfloor R/N_R \rfloor + \lfloor \alpha/N_\alpha \rfloor + \lfloor z/N_z \rfloor \\ R = \sqrt{(x-x_c)^2 + (y-y_c)^2}\ ;\ \alpha = tan^{-1}\left(\frac{y-y_c}{x-x_c}\right) \end{gathered} \tag{8}$$

Similarly, for spherical checkers array, (7b) becomes

$$\begin{gathered} u \triangleq \left\lfloor \frac{R}{N_R} \right\rfloor + \left\lfloor \frac{\alpha}{N_\alpha} \right\rfloor + \left\lfloor \frac{\beta}{N_\beta} \right\rfloor \\ R = \sqrt{(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2} \\ \alpha = tan^{-1}\left(\frac{y-y_c}{x-x_c}\right);\ \beta = sin^{-1}\left(\frac{z-z_c}{R}\right) \end{gathered} \tag{9}$$

where x_c, y_c, z_c (in pixels) refer to the origin of the coordinate system while N_R (pixels), N_α (radians), N_β (radians) refer to the parameters governing the size of the transformed cells. Besides checkers, an unlimited number of textured materials can be implemented/modelled using elementary mathematical transforms and mapping techniques. Consequently, applying the erosion operator (when Φ = 1) along the contours of patterns in each and every image in **S** yields patterned indentations (with a depth of r pixels). Fig. 1 illustrates the flow chart of the algorithm.

Note that in this simplified scheme, the erosion parameter r directly controls the depth of the indentations where the characteristic size of the features (i.e. cell size) must be greater than 2r. As a remedy, the following operation could be utilized to obtain finer indentations:

$$\mathbf{I}_k \coloneqq \mathbf{I}_k \wedge (\mathbf{I_0} \wedge \mathbf{\Phi_k}), \qquad \forall k \in \mathbb{N}_{\leq K} \tag{10}$$

where $\mathbf{I_0}$ refers to the eroded $\mathbf{I_k}$ (by r pixels) while $\mathbf{\Phi_k} \in \mathbb{B}^{N\times M}$ can be expressed as

$$\mathbf{\Phi_k} = [\Phi(x,y,z)]_{z=kL_z}, \forall x \in \mathbb{N}_{\leq N}, \forall y \in \mathbb{N}_{\leq M} \tag{11}$$

Despite its simplicity, this (computationally costly) approach might lead to certain problems in FDM type of 3D printing.

It is critical to notice that the proposed method directly employs a sequence of binary images. If the resolution of the printed artifact is relatively high, (just like STL files) the storage requirement could be significant (>> 1 GB). Consequently, the image compression as suggested by [21] is a complementary component of the methodology.

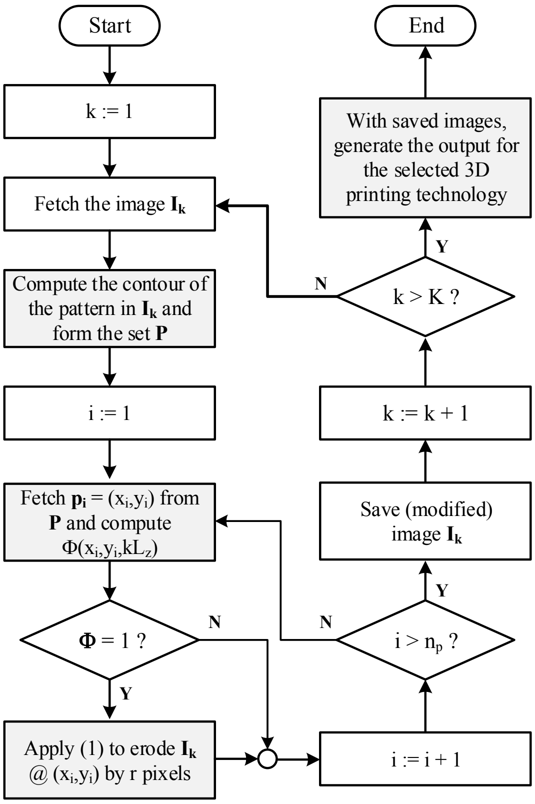

Start
k := 1
Fetch the image $\mathbf{I_k}$
Compute the contour of the pattern in $\mathbf{I_k}$ and form the set **P**
i := 1
Fetch $\mathbf{p_i} = (x_i,y_i)$ from **P** and compute $\Phi(x_i,y_i,kL_z)$
Φ = 1 ?
N
Y
Apply (1) to erode $\mathbf{I_k}$ @ (x_i,y_i) by r pixels
i := i + 1
i > n_p ?
N
Y
Save (modified) image $\mathbf{I_k}$
k := k + 1
k > K ?
N
Y
With saved images, generate the output for the selected 3D printing technology
End

Fig. 1. Flow chart of the proposed method.

C. Post-processing

Once the image sequences defining the object with patterned indentations are formed, the next step is to create printer commands suitable for the selected AM technology. It is self-evident that the approach presented here is directly compatible with DLP printers where binary images are sent to a projector curing photopolymers to produce the artifact slice by slice. Hence, for DLP printers, the modified images (as discussed in Section B) can be transferred in sequential order at predefined time intervals. The number of images depends on the defined layer thickness, which is a direct consequence of utilized photopolymer and the light source.

On the other hand, FDM printers do pose a greater challenge since the path of the extruder head inside each and every cross-section must be defined along with certain process parameters (such as extruder temperature, filament feed speed, extruder clearance, etc.). The generated output is usually a special NC-code (abiding RS-274D). The code, by and large, represents the extruder trajectory as polylines/polygons (i.e. a sequence of G0/G1 codes). Note that the motion controller of the printer processes the code to generate commands to the corresponding units of the printer (i.e stepper motor drivers, extruder heater, etc.) To that end, a trajectory planning must be performed.

For this purpose, morphological operations on binary images (as elaborated in Section A) are again facilitated. For a particular binary image, the erosion operator, whose parameter r is to be interpreted as the distance between two successive tool paths, is applied along the contour of the pattern in the image. Using boundary tracing technique, the contour of the eroded image (to be converted to the parameters of G1 code at the latter stages) is extracted. The above-mentioned procedure is repeated several times until no white pixels left in the image that is modified in succession. Fig. 2 shows the flow chart of the algorithm where ~ sign denotes the modified bitmap images containing the patterned indentations. Similarly, Fig. 3 illustrates the parallel paths generated for a generic case (i.e. a crosssection of the Stanford Bunny in Section IV).

Notice that the optimal passage between routes is determined using the nearest neighbor technique. When the distance between two points exceeds a specific threshold value (i.e. 2r), the following steps are applied: **i)** the flow through the extruder is stopped; **ii)** the extruder head is elevated to a safe distance; **iii)** the head moves on a recti-linear path to reach the destination point at rapid-travel speed (which is latter to be translated to G0 code); **iv)** the extruder head moves down to the printing depth; **v)** printing resumes.

IV. Implementation & Test Results

As an illustration of the proposed methods, two cases are considered: **i)** Vase [22]; **ii)** Stanford Bunny [23]. Fig. 4 shows rendered images of the STL models while Table I summarizes the important attributes for both cases to be fabricated via DLP- and FDM processes. For this purpose, the bitmap images of the cross-sections are obtained via the utility software of B9Creator DLP printer and the algorithms discussed in Section III are all implemented in MATLAB 2016b. Notice that the proposed methods lend themselves for easy implementation on DLP printers where the processed images are simply imported into the available printing file (in B9J format) to fabricate the parts.

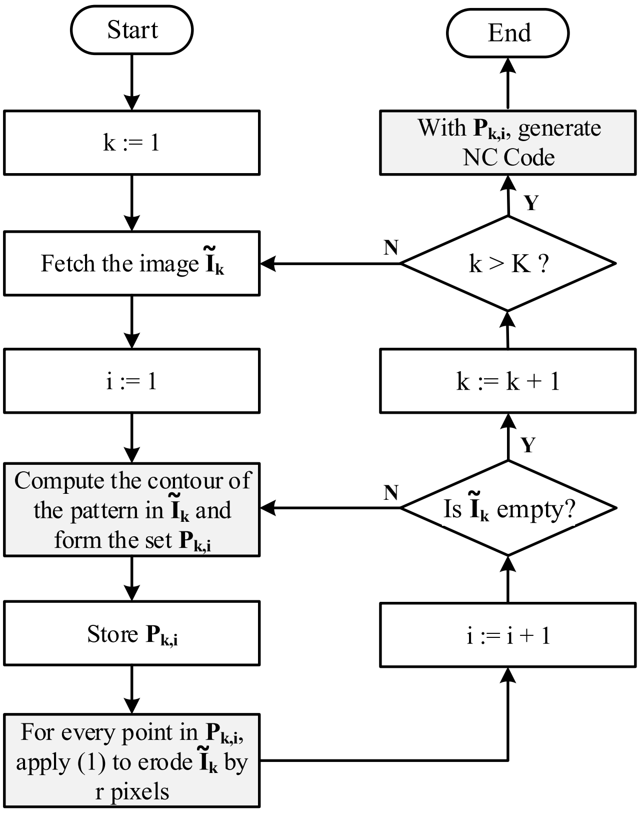

Start
k := 1
Fetch the image $\tilde{\mathbf{I}}_k$
i := 1
Compute the contour of the pattern in $\tilde{\mathbf{I}}_k$ and form the set $\mathbf{P}_{k,i}$
Store $\mathbf{P}_{k,i}$
For every point in $\mathbf{P}_{k,i}$, apply (1) to erode $\tilde{\mathbf{I}}_k$ by r pixels
i := i + 1
Is $\tilde{\mathbf{I}}_k$ empty?
N
Y
k := k + 1
k > K ?
N
Y
With $\mathbf{P}_{k,i}$, generate NC Code
End

Fig. 2. Simplified flow chart of the post-processor.

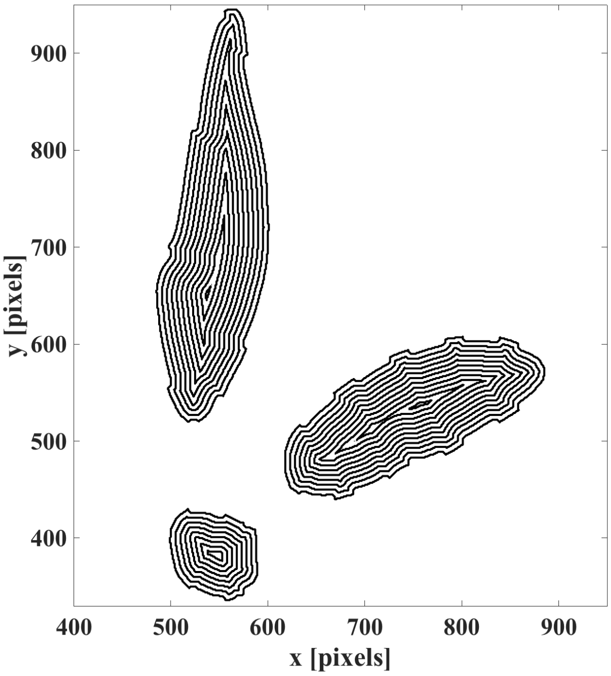
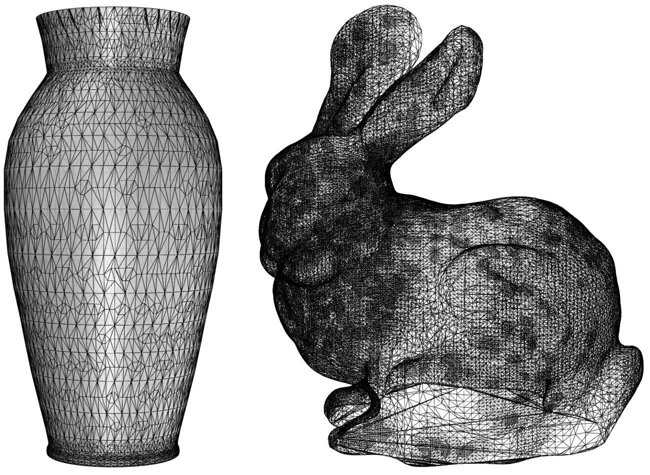


Fig. 3. Parallel paths (curves) generated by the presented algorithm



a) Vase *b) Stanford Bunny*

Fig. 4. Rendered images of the STL models used in the test cases.

TABLE I. ATTRIBUTES OF THE TEST CASES

	Vase	Stanford Bunny
DLP Print Size [mm]	33.99 × 33.99 × 68.18	43.15 × 33.43 × 41.87
FDM Print Size [mm]	24.15 × 24.15 × 72.41	86.29 × 66.87 × 83.75
STL File Size [KB]	911	4230
DLP Slice Thickness	70 μm	
FDM Slice Thickness	60 μm	
Image Size	1920 × 1080 pixels	

On the other hand, FDM process embodies a number of technical challenges. That is, the developed programs (i.e. M-scripts) accepting binary bitmap images as inputs generate the "indented" images for each and every layer depending on the texture function selected. Contours of the modified images, which are represented as 2D-polygons, are then obtained by boundary tracing methods. The post-processor specifically devised for an Ultimaker 2 Go 3D printer employs the vertices of these polygons to generate the corresponding NC code. Table II tabulates the FDM printing parameters. To minimize the fabrication time as well as the amount of deposited material, only the outer shells (with 1.2 mm thickness) of the artifacts are fabricated.

Notice that for the Vase case, the spherical checkers array is selected while the origin of the coordinate system is set as the bottom center of the model. With respect to the Stanford Bunny, the special form of (7), which leads to diamond patterns on the surface, is facilitated:

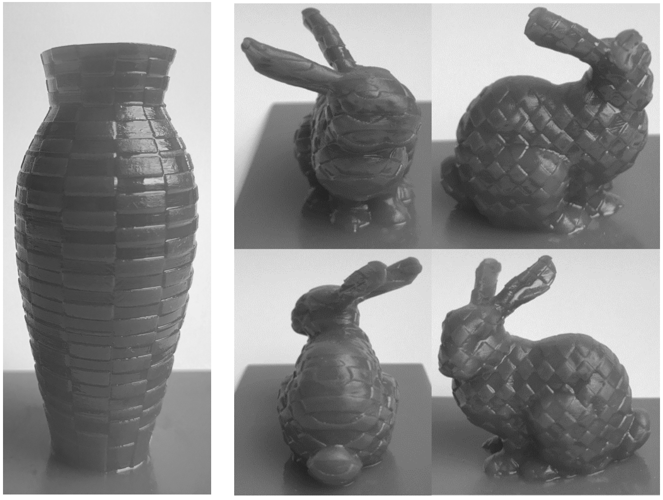
$$\Phi(x,z) = \begin{cases} 1, & u = 2\left\lfloor \frac{u}{2} \right\rfloor \\ 0, & else \end{cases} \tag{12a}$$

$$\begin{gathered} u \triangleq \lfloor x'/N_x \rfloor + \lfloor z'/N_z \rfloor \\ \begin{bmatrix} x' \\ z' \end{bmatrix} = \begin{bmatrix} c\theta_y & s\theta_y \\ -s\theta_y & c\theta_y \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \end{gathered} \tag{12b}$$

where $\theta_y = \pi/4$.

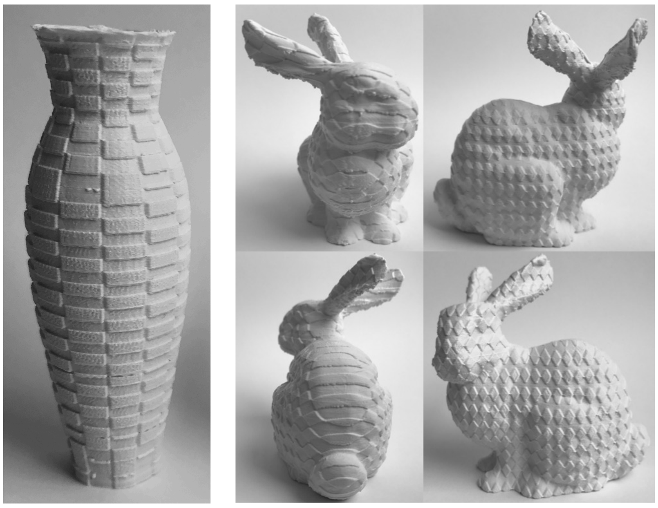
Figs. 5 and 6 illustrate the fabricated objects via DLP and FDM processes, respectively. Notice that large flat surfaces at the bottom of the objects (i.e. "base") in Fig. 5 are used to increase the contact area between the photopolymer (titled B9R-1-RED) and the build table.

As can be seen, despite some minor glitches, the desired (intended) patterns on the surfaces of the objects are successfully attained without the utilization of CAD software (or the input/intervention of the user). It is critical to note that Eqn. (7a), by design, is susceptible to quantization noise which is induced by traced boundaries of binary images. The irregular regions on the Vase (see Figs. 5a and 6a) are actually the manifestations of that effect. One remedy for this problem is the use of low-pass filters to smooth out the above-mentioned paths.



a) Vase *b) Stanford Bunny*

Fig. 5. Fabricated objects using a DLP printer.



c) Vase *d) Stanford Bunny*

Fig. 6. Fabricated objects using an FDM printer.

TABLE II. PRINTING PARAMETERS OF THE FDM PROCESS

Parameter	Value
Material	PLA
Filament Diameter	2.85 mm
Nozzle Diameter	0.4 mm
Extrusion Temperature	210 °C
Layer Thickness	70 µm
Printing Speed	2000 mm/min
Non-printing Speed	9000 mm/min
Shell Thickness	1.2 mm

V. CONCLUSIONS

This paper presented an easy-to-implement technique to create textured indentations on 3D printed parts. The method leverages the morphological operations on binary images representing the cross-sections of the printed artefacts. Due to its utter simplicity, the proposed technique can easily be implemented on printer hardware without the need for the intervention of the CAD software. Another major advantage of the approach is that once it is realized on any type of AM machinery, various artifacts can be manufactured with the same type of surface patterns provided that the input images are available. Although the method was not realized completely on the hardware of 3D printers, its applicability has been assessed on 3D printers employing different technologies (DLP and FDM).

As a future work, the method is planned to be implemented on a single board computer (most likely on Raspberry Pi 3), which will drive two different printers (DLP and FDM). The user will be able to switch easily from one technology to the other via setting some parameters on the manufacturing file.

REFERENCES

[1] H. Jee and E. Sachs, "Surface macro-texture design for rapid prototyping," Rapid Prototyping Journal, vol. 6, no. 1, pp. 50-60, 2000.

[2] I. Gibson, L.K. Cheung, S.P. Chow, W.L. Cheung, S.L. Beh, M. Savalani, and S.H. Lee, "The use of rapid prototyping to assist medical applications," Rapid Prototyping Journal, vol. 12, no. 1, pp. 53-58, 2006.

[3] M. Livesu, S. Ellero, J. Martínez, S. Lefebvre, and M. Attene, "From 3D models to 3D prints: an overview of the processing pipeline," Computer Graphics Forum, vol. 36, no. 2, pp. 537-564, 2017.

[4] U. Yaman and M. Dolen. "A command generation approach for desktop fused filament fabrication 3D printers," 42nd Annual Conference of the IEEE - Industrial Electronics Society, IECON, pp. 4588-4593, 2016.

[5] A. Armillotta, "Assessment of surface quality on textured FDM prototypes," Rapid Prototyping Journal, vol. 12, no. 1, pp. 35-41, 2006.

[6] A.H. Bermano, T. Funkhouser, and S. Rusinkiewicz, "State of the Art in Methods and Representations for Fabrication-Aware Design," Computer Graphics Forum, vol. 36, no. 2, pp. 509-535, 2017.

[7] D. Panozzo, O. Diamanti, S. Paris, M. Tarini, E. Sorkine, and O. Sorkine-Hornung, "Texture Mapping Real-World Objects with Hydrographics," Computer Graphics Forum, vol. 34, no. 5, pp. 65-75, 2015.

[8] C. Schüller, D. Panozzo, A. Grundhöfer, H. Zimmer, E. Sorkine, and O. Sorkine-Hornung, "Computational Thermoforming," ACM Transactions on Graphics (TOG), vol. 35, no. 4, article no. 43, 2016.

[9] E. Pei, "4D Printing: dawn of an emerging technology cycle," Assembly Automation, vol. 34, no. 4, pp. 310-314, 2014.

[10] K. Vidimče, S.P. Wang, J. Ragan-Kelley, and W. Matusik, "OpenFab: a programmable pipeline for multi-material fabrication," ACM Transactions on Graphics (TOG), vol. 32, no. 4, article no. 136, 2013.

[11] H. Takahashi and H. Miyashita, "Expressive Fused Deposition Modeling by Controlling Extruder Height and Extrusion Amount," Proceedings of the Conference on Human Factors in Computing Systems, pp. 5065-5074, 2017.

[12] Y. Kanada, "3D printing of generative art using the assembly and deformation of direction-specified parts," Rapid Prototyping Journal, vol. 22, no. 4, pp. 636-644, 2016.

[13] O. van Herpt, "Solid Vibraitons," http://oliviervanherpt.com/solid-vibrations, accessed: 2017-06-06.

[14] G. Laput, X.A. Chen, and C. Harrison, "3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles," The 28th Annual ACM Symposium on User Interface Software & Technology, pp. 593-597, 2015.

[15] J. Ou, G. Dublon, C.Y. Cheng, F. Heibeck, K. Willis, and H. Ishii, "Cilllia: 3D Printed Micro-Pillar Structures for Surface Texture, Actuation and Sensing," CHI Conference on Human Factors in Computing Systems, pp. 5753-5764, 2016.

[16] T. Reiner, N. Carr, R. Měch, O. Šťava, C. Dachsbacher, and G. Miller, "Dual-color mixing for fused deposition modeling printers," Computer Graphics Forum, vol. 33, no. 2, pp. 479-486, 2014.

[17] A. Brunton, C.A. Arikan, and P. Urban, "Pushing the limits of 3d color printing: Error diffusion with translucent materials," ACM Transactions on Graphics (TOG), vol. 35, no. 1, article no. 4, 2015.

[18] J. Serra, Image analysis and mathematical morphology, Academic, London, 1982.

[19] T.L. Chia, K.B. Wang, L.R. Chen, and Z. Chen, "A parallel algorithm for generating chain code of objects in binary images," Information Sciences vol. 149, no. 4, pp. 219-234, 2003.

[20] E. Angel and D. Shreiner, Interactive Computer Graphics with WebGL, Addison-Wesley Professional, 2014.

[21] U. Yaman, M. Dolen, U.M. Dilberoglu and B. Gharehpapagh, "A new method for generating image projections in DLP-type 3D printer systems," Proceedings of 27th International Conference on Flexible Automation and Intelligent Manufacturing, 2017.

[22] Vase, https://www.turbosquid.com/3d-models/free-3ds-mode-kartell-misses-flower-power/1025377, accessed: 2017-06-06.

[23] Stanford Bunny, https://graphics.stanford.edu/data/3Dscanrep, accessed: 2017-06-06.