IISE Transactions

ISSN: 2472-5854 (Print) 2472-5862 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

Generation of patterned indentations for additive manufacturing technologies

Ulas Yaman, Melik Dolen & Christoph Hoffmann

To cite this article: Ulas Yaman, Melik Dolen & Christoph Hoffmann (2019) Generation of patterned indentations for additive manufacturing technologies, IISE Transactions, 51:2, 209-217, DOI: 10.1080/24725854.2018.1491076

To link to this article: https://doi.org/10.1080/24725854.2018.1491076

Accepted author version posted online: 28 Jun 2018.
Published online: 25 Sep 2018.

IISE TRANSACTIONS
2019, VOL. 51, NO. 2, 209–217
https://doi.org/10.1080/24725854.2018.1491076

Taylor & Francis
Taylor & Francis Group

Generation of patterned indentations for additive manufacturing technologies

Ulas Yaman[a], Melik Dolen[a], and Christoph Hoffmann[b]

[a]Department of Mechanical Engineering, Middle East Technical University, Ankara, Turkey; [b]Department of Computer Science, Purdue University, West Lafayette, IN, USA

ABSTRACT

This article proposes a novel approach to generate patterned indentations for different additive manufacturing methodologies. Surface textures have many practical applications in various fields, but require special manufacturing considerations. In addition to 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 that respect. For instance, the size of the design (CAD) files grows considerably when there are detailed indentations on the surfaces of the artifacts. The presented method, which employs morphological operations on a sequence of binary images representing the cross-sections of the printed artefact, overcomes such problems while fabricating the textured objects. Furthermore, the presented technique could be conveniently implemented using the existing hardware resources of almost any three-dimensional printer.

ARTICLE HISTORY
Received 29 November 2017
Accepted 8 June 2018

KEYWORDS
Surface textures; indentation; 3D printing; additive manufacturing; FDM

1. Introduction

Surface textures are the geometric features formed on surfaces of artefacts (Jee and Sachs, 2000). They are used for a multitude of purposes in industry and academia. For instance, in the mechanical engineering domain, they can be utilized:

- to increase surface areas for better heat transfer,
- to attain favorable tribological properties (Jee and Sachs, 2000),
- 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 (Van Rompay *et al.*, 2017),
- to imitate the surface properties of actual objects (like bones to design orthopedic implants (Gibson *et al.*, 2006)).

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 (Yaman and Dolen, 2016; Livesu *et al.*, 2017), 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 frequently 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, commercial CAD packages do not offer any utilities/tools to create the patterned indentations discussed in this article. 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 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. Specifically, the texture is represented by a sequence of binary bitmap images that can be manipulated by fast and computationally-inexpensive image processing algorithms. Since the Boolean operations on binary images are not computationally costly, the desired texture properties can be easily incorporated into the images on the fly (even inside the printer), using cheap commodity PCs with limited resources.

CONTACT Ulas Yaman uyaman@metu.edu.tr
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uiie.

The details of the proposed method are discussed in the following sections, with the effectiveness of the presented method being demonstrated through two test cases using a Fused Deposition Modeling (FDM) and a Digital Light Processing (DLP) type of three-dimensional (3D) printing technologies.

The rest of this article 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 article concludes with some key remarks and discussions for future directions of work.

2. Background

The practical importance of surface textures has motivated secondary manufacturing methods to create these textures on the surfaces of artifacts, such as painting, coating, engraving, etc. In terms of surface textures, the major advantage of AM methods over conventional manufacturing operations is that patterns can be potentially realized without secondary operations (Armillotta, 2006). Hence, AM is preferred for manufacturing complex geometric patterns (Bermano *et al.*, 2017). For 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 (Panozzo *et al.*, 2015) and computational thermoforming (Schüller *et al.*, 2016).

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 to extend the capabilities of AM machinery (Pei, 2014). One of the major drawbacks is the geometric deviation of the fabricated parts from the designed models. In a recent study, Cheng *et al.* (2018) proposed a method to predict the deviation based on the process parameters of FDM and to update the initial design file accordingly. In another approach, Vidimce *et al.* (2013) 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 (2017) have drawn patterns by adjusting the two main printing parameters: (i) clearance between the extruder tip and the corresponding layer; and (ii) the amount of extruded material. For embossing letters on vertical walls, they simply adjusted the extrusion rate. This approach resembles the one presented by Kanada and Kanada (2016). The major difference is that in the study of Kanada and Kanada (2016) the patterns are generated via a spiral/helical printing technique, in which the whole path is seamless. A recent patent by Mayer (2016) also focuses on the extrusion rate to obtain surface textures. He claims that resolutions of the textures can be better than the resolution of the corresponding 3D printer when the parameters are set properly. A totally different method to yield surface patterns on FDM printers was developed by Van Herpt (2015). In this technique, the printer is excited from its base and there is no other modification of 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.* (2015) created hair-like structures by modifying the parameters of the extrusion process. Likewise, Ou *et al.* (2016) 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.* (2014) 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. There are also studies for using multi-extruders simultaneously in order to decrease the fabrication time of the artifacts (Jin *et al.*, 2017). These studies can be extended to obtain more complicated textures and colored prints (Hergel and Lefebvre, 2014). 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.* (2015) expanded the capabilities of multi-jet printers by employing an error diffusion approach. They 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 overarching solution for DLP-type 3D printers. In this study, we propose a method especially suitable for DLP printers. However, it can also be adapted to other AM technologies (like FDM, stereolithography) by employing appropriate post-processing algorithms accommodating the requirements of a particular technology (as discussed and realized in the upcoming sections).

3. Proposed method

This article 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. The pipeline of the proposed method is summarized in Figure 1. As can be seen, after the slices of the corresponding CAD model are obtained, they are stored as binary images. Consequently, morphological operations on these images are performed to produce desired patterns, based on the parameters set by the user. If the chosen AM method is different from DLP printing, some post-processing is required to fabricate the model with the selected patterns. For instance, for FDM-type printers,

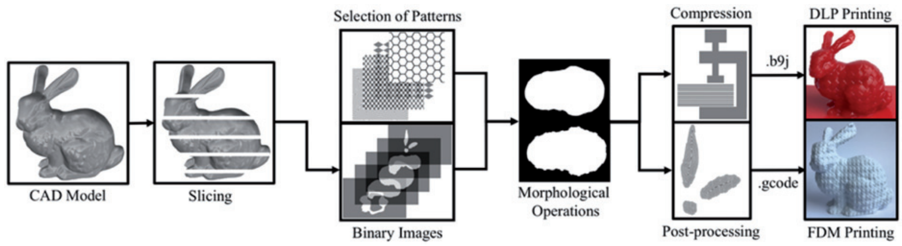

Figure 1. Pipeline of the proposed method.

the transformed binary images are further processed to get the extruder trajectories for each and every cross-section of the artifact. These trajectories can then be converted into a standard NC code ("gcode"), which is subsequently interpreted on the controller of the printer. For DLP technology, the modified binary images are simply compressed so that the controller can extract a particular binary image for projection at any time. The critical parts of this pipeline are discussed in the following sub-sections.

3.1. 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 (Serra, 1982).

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 $\mathbf{p} = (x, y) \in \mathbb{N}^2$ in the image sets the bits to zero inside the circular region centered at p with radius $r \in \mathbb{N}$:

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

where $\mathbf{I}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is the sub-matrix (sub-image) of $\mathbf{I}$ 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:

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

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

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

Similarly, the dilation operation can be expressed as follows:

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

Hence, one can apply the Boolean operations defined in Equations (1) and (4) along the contours of the pattern(s) in $\mathbf{I}$ to obtain the desired transformation. To that end, boundary tracing algorithms (Chia *et al.*, 2003) can be employed to find the pixel locations at the outer periphery of the pattern(s) contained in the image:

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

where n_p denotes the number of points on the boundary. When the operation in Equation (4) (or Equation (5)) is conducted for all the points in Equation (5), the pattern(s) in $\mathbf{I}$ is simply eroded (or dilated) by r pixels.

3.2. 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 $\mathbf{S}$ be a set of images representing the successive cross-sections of a 3D printed object:

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

where K is the number of images in this set which are set by L_z (i.e., the layer thickness) pixels apart For this purpose, a texture function $\Phi(x,y,z)$ ($\Phi:\mathbb{N}^3 \rightarrow \mathbb{B}$) needs to be defined to indicate the locations of the features in the image space (or the domain). If $\Phi(x,y,z)$ returns one (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 (Angel and Shreiner, 2014) could be utilized for the applications where a two-dimensional (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:

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

$$u \triangleq \lfloor x'/N_x \rfloor + \lfloor y'/N_y \rfloor + \lfloor z'/N_z \rfloor, \tag{7b}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c\theta_y c\theta_z & -c\theta_y s\theta_z & s\theta_y \\ s\theta_x s\theta_y c\theta_z + c\theta_x s\theta_z & -s\theta_x s\theta_y s\theta_z + c\theta_x c\theta_z & -s\theta_x c\theta_y \\ -c\theta_x s\theta_y c\theta_z + s\theta_x s\theta_z & c\theta_x s\theta_y s\theta_z + s\theta_x c\theta_z & c\theta_x c\theta_y \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{7c}$$

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); θ_x, θ_y, and θ_z refer to the rotations around the major axes. Notice that Equation (7a) is simply a Boolean function for testing the parity (i.e., oddness/evenness) of its integer argument (u) whereas u of Equation (7b) can be regarded as the summation of indices addressing the location of a particular cell in the checkers array. Since u only changes by ±1 between two neighboring cells, one can conveniently create alternating (binary) patterns like a 3D checkerboard. Similarly, Equation (7c) yields a coordinate transformation between the main reference frame $\{x,y,z\}$ and the one rotated around principle axes (i.e., $\{x',y',z'\}$). Hence, changing the angles θ_x, θ_y, and θ_z enables users to control the orientation of the 3D checkerboard arbitrarily.

Other arrays could be formed using different coordinate systems. In a cylindrical coordinate system, a "polar" checkers array could be easily produced. Hence, Equation (7b) takes the following form:

$$u \triangleq \lfloor R/N_R \rfloor + \lfloor \alpha/N_\alpha \rfloor + \lfloor z/N_z \rfloor,$$

$$R = \sqrt{(x-x_c)^2 + (y-y_c)^2} \;\; ; \;\; \alpha = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right). \tag{8}$$

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

$$u \triangleq \left\lfloor \frac{R}{N_R} \right\rfloor + \left\lfloor \frac{\alpha}{N_\alpha} \right\rfloor + \left\lfloor \frac{\beta}{N_\beta} \right\rfloor ,$$

$$R = \sqrt{(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2} ,$$

$$\alpha = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right); \;\; \beta = \sin^{-1}\left(\frac{z-z_c}{R}\right) , \tag{9}$$

where x_c, y_c, z_c (in pixels) refer to the origin of the coordinate system, and N_R (pixels), N_α (radians), N_β (radians) refer to the parameters governing the size of the transformed cells. In addition to checkers, an unlimited number of textured materials can be implemented/modelled using elementary mathematical transforms and mapping techniques. Consequently, applying the erosion operator (when $\Phi = 1$) along the contours of patterns in each and every image in **S** yields patterned indentations (with a depth of r pixels). Figure 2 illustrates the flow chart of the algorithm.

Note that in this scheme, the r-parameter of the morphological operator essentially controls the depth of the indentations. The minimum indentation depth is obviously one pixel. However, the maximum depth (i.e., the upper bound of r) is directly related to the minimum cell size of the indented pattern (L_{min}) since the structuring element **T** (see Equation (2)) erodes circular regions around the periphery of the indented portions. Hence, the depth of the indented pattern must be less than half of L_{min}. As a remedy, the following strategy could be adapted to obtain finer indentations:

$$\mathbf{I}_k := \mathbf{I}_k \vee (\mathbf{I}_0 \wedge \mathbf{\Phi}_\mathbf{k}), \;\; \forall k \in \mathbb{N}_{\leq K} \;\; , \tag{10}$$

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

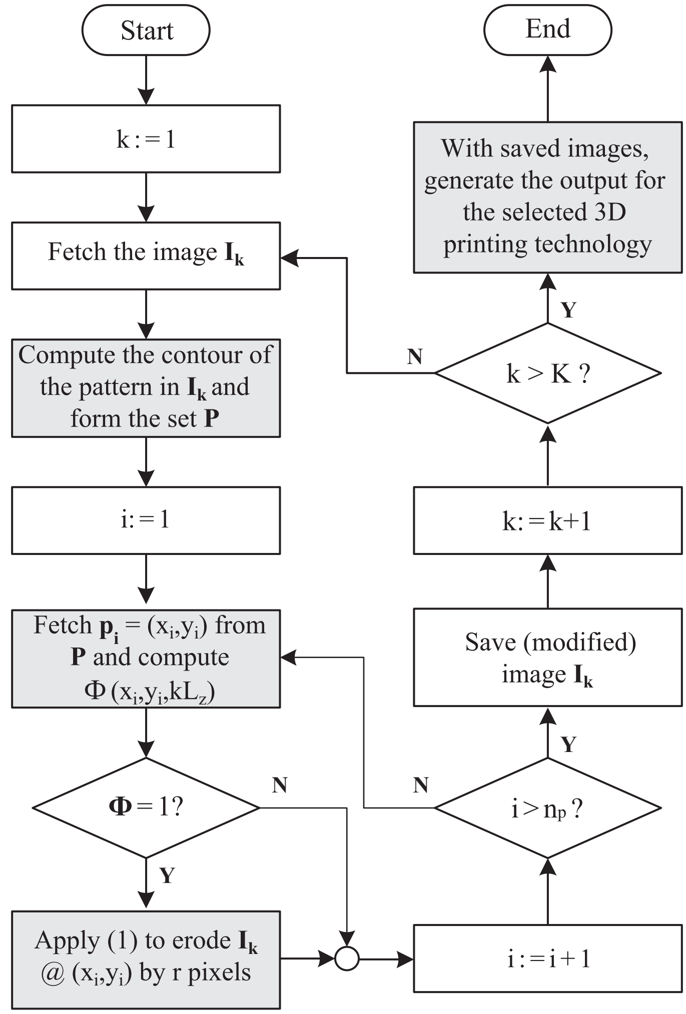

Figure 2. Flow chart of the proposed method.

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

Figure 3 illustrates a sample operation indicated by Equation (10). As can be seen from Figure 3(d), the underlying procedure yields sharp indentations if compared with its counterpart. Despite its simplicity, the application of Equation (10) might lead to certain problems in FDM printing. Figure 4 shows some common defects resulting from this alternative technique; point (A):Voids near the surface, point (B): sharp protrusions, point (C): orphan indentations, and point (D): abrupt changes on the surface texture that could not be effectively fabricated via FDM technology. Note that the former method eliminates all of the above-mentioned defects by rounding the corners of the indentations. Thus, the method inherently incorporates a low-pass filter to smooth out the sharp changes registered on the surface as an intrinsic property of its erosion operator.

One possible solution to obtain finer indentations with fewer flaws is to use an adaptive morphological operator that takes into account not only the position of the operator, but also the gradient of the traced curve at that particular location. That is, the shape of the structuring element is adaptively modified accordingly. However, the evaluation of this new approach is left open for future studies.

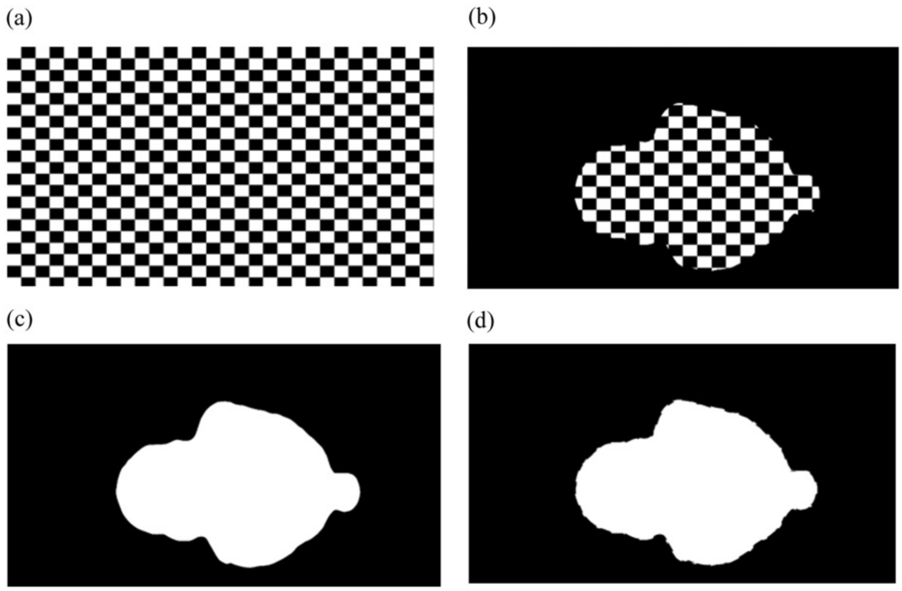

Figure 3. Operations for finer indentations: (a) binary image Φ_k; (b) binary image formed by $I_k \wedge \Phi_k$; (c) eroded image around its contours: I_0; and (d) composite image: $I_0 \vee (I_k \wedge \Phi_k)$.

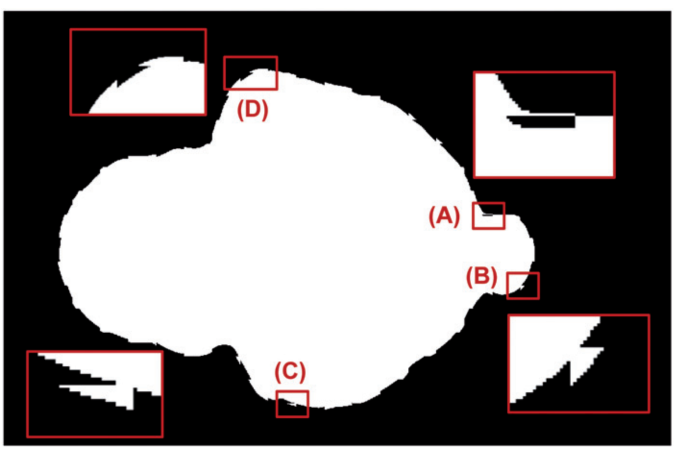

Figure 4. Potential defects while producing finer indentations.

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 ($\gg$ 1 GB). Consequently, the image compression as suggested by Yaman *et al.* (2017) is a complementary component of the methodology.

3.3. 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 3.2) 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 the photopolymer and the light source used.

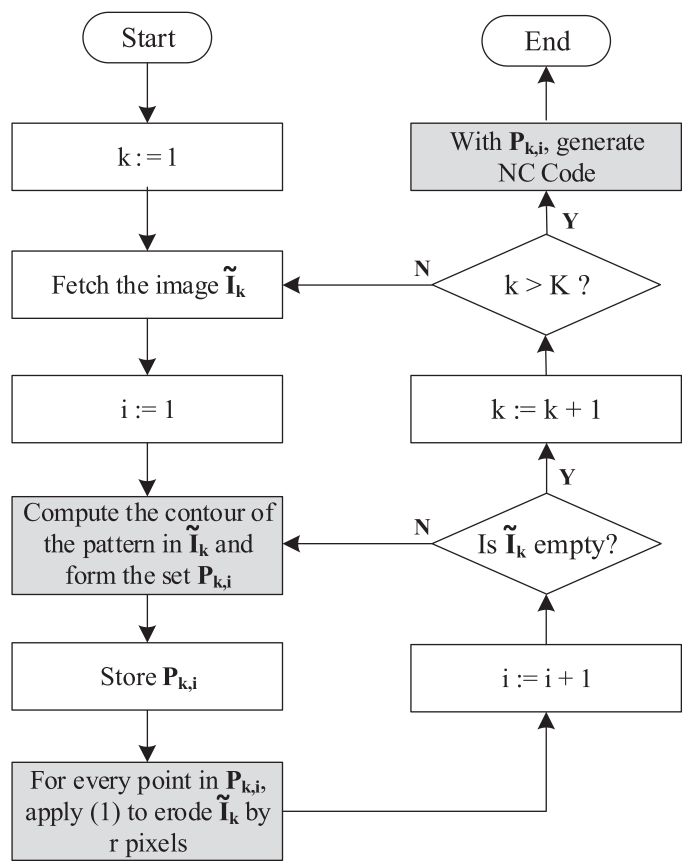

Figure 5. Simplified flow chart of the post-processor.

On the other hand, FDM printers do pose a greater challenge, as 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 (conforming with RS-274D). The code, by and large, represents the extruder trajectory as polylines/polygons (i.e., a sequence of G0/G1 codes).

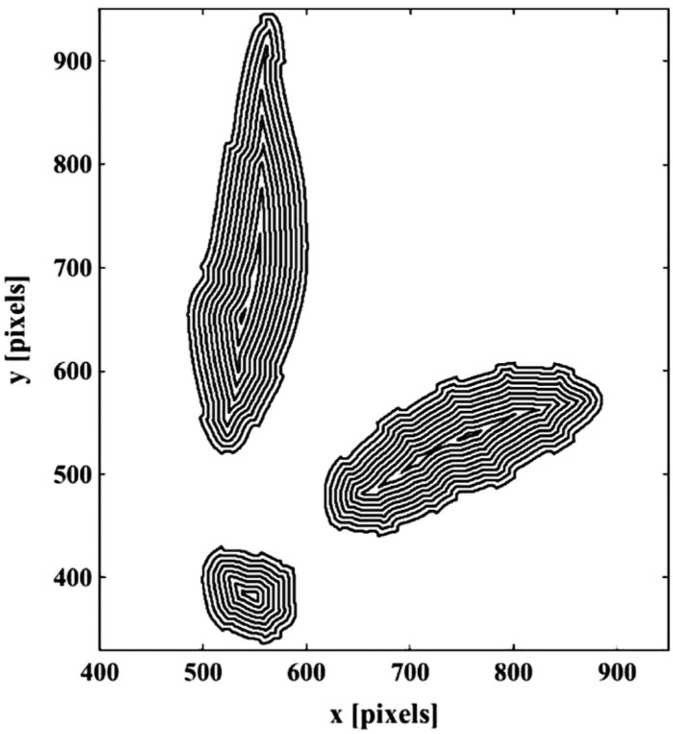

Figure 6. Parallel paths (curves) generated by the presented algorithm.

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, trajectory planning must be performed.

For this purpose, morphological operations on binary images (as elaborated in the previous sub-section) are again facilitated. For a particular binary image, the erosion operator is applied along the contour of the pattern in the image. The parameter r of the operator is interpreted as the distance between two successive tool paths. Using a 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 that changed in the last pass are left in the image. Figure 5 shows the flow chart of the algorithm. The ~ sign denotes the modified bitmap images containing the patterned indentations. Similarly, Figure 6 illustrates the parallel paths generated for a generic case (i.e., a cross-section of the Stanford Bunny in the upcoming Section).

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 a rapid travel speed (which is later to be translated to G0 code); (iv) the extruder head moves down to the printing depth; (v) printing resumes.

4. Implementation & test results

As an illustration of the proposed methods, two cases are considered: (i) Vase (Vase, 2018); and (ii) Stanford Bunny (Stanford Bunny, 2018). Figure 7 shows rendered images of the STL models while Table 1 summarizes the important attributes for both cases to be fabricated via DLP or FDM processes. For this purpose, the bitmap images of the cross-sections are obtained via the utility software of the B9Creator DLP printer and the algorithms discussed in the previous section are all implemented in MATLAB 2016b. Notice that the proposed methods are easy to implement on DLP printers where the processed images are simply imported into the available printing file (in B9J format) to fabricate the parts.

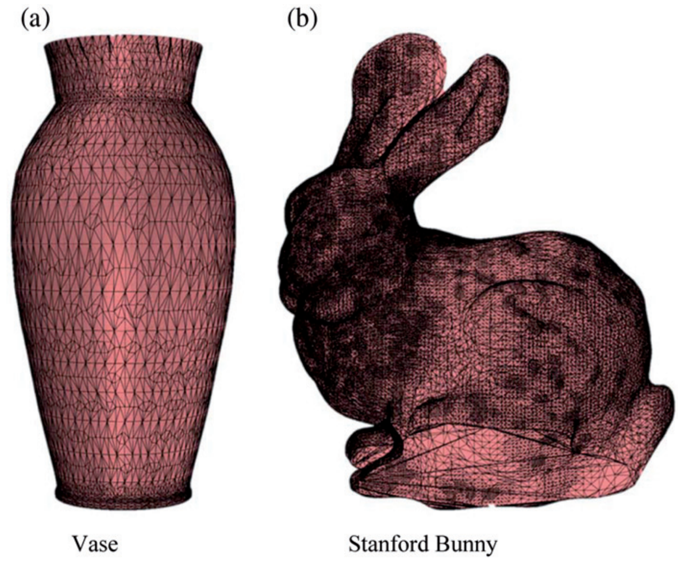

Figure 7. Rendered images of the STL models used in the test cases.

On the other hand, the 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 2 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. For the Stanford Bunny, the special form of Equation (7) is helpful and leads to diamond patterns on the surface:

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

$$u \triangleq \left\lfloor \frac{x'}{N_x} \right\rfloor + \left\lfloor \frac{z'}{N_z} \right\rfloor,$$

Table 1. Attributes of the test cases.

	Vase	Stanford bunny
DLP print size (mm)	$33.99 \times 33.99 \times 68.18$	$43.15 \times 33.43 \times 41.87$
FDM print size (mm)	$24.15 \times 24.15 \times 72.41$	$86.29 \times 66.87 \times 83.75$
STL file size (kB)	911	4230
DLP slice thickness (μm)	70	
FDM slice thickness (μm)	60	
Image size (pixels)	1920×1080	

Table 2. 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	60 μm
Printing speed	2000 mm/min
Non-printing speed	9000 mm/min
Shell thickness	1.2 mm

$$\begin{bmatrix} x' \\ z' \end{bmatrix} = \begin{bmatrix} c\theta_y & s\theta_y \\ -s\theta_y & c\theta_y \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}, \tag{12b}$$

where $\theta_y = \pi/4$.

Figures 8 and 9 illustrate the fabricated objects via the DLP and FDM process, respectively. Notice that large flat surfaces at the bottom of the objects (i.e., "base") in Figure 8 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 patterns on the surfaces of the objects are successfully obtained without using the CAD software (or any input/intervention by the user). In fact, the observed defects/glitches on the printed objects can be categorized into two groups: (i) computational/numerical representation errors (such as rounding/truncation/quantization errors); and (ii) AM (technology)-induced errors. Figure 10 demonstrate some of the above-mentioned defects in the Stanford Bunny case. For instance, Figures 10(a) and 10(c) show that the textured patterns are discontinuous at the edges, due to the fact that the small features to be created on binary images become subpixel in size. Hence, they disappear as a consequence of the rounding errors in the relevant computations (see Equation (12)). On the other hand, the properties of the printer (i.e., printing technology) along with the material quality manifest themselves as AM-induced errors, such as the ones presented in Figures 10(b) and 10(d). In Figure 10(b), upward fluctuation in the extrusion rate results in an excessive amount of deposited material on that particular layer. Hence, it creates the bulging effect that appears to be a misaligned layer. Similarly, in DLP printing, some excess material (i.e., uncured photopolymer) left in shallow regions hinders small features, so that a shiny surface emerges, as shown in Figure 10(d). Similar deficiencies in the Vase case can be detected from Figures 8(a) and 9(a).

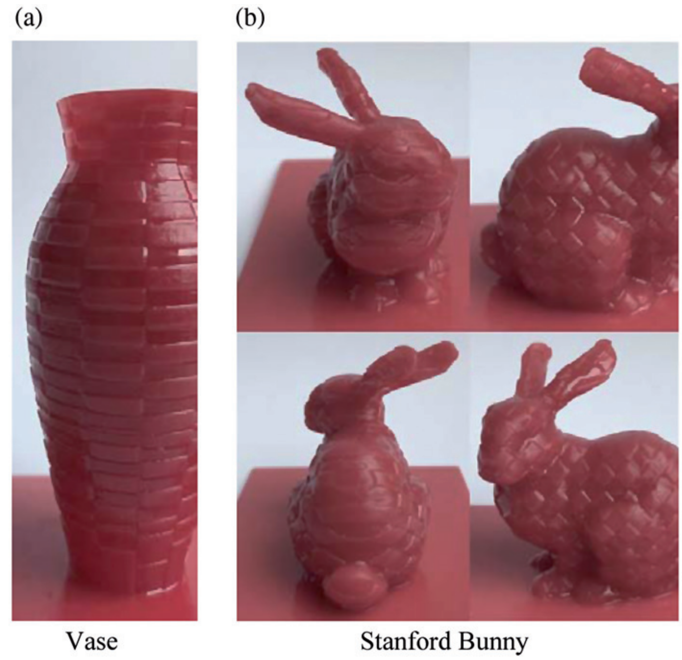

Figure 8. Fabricated objects using a DLP printer.

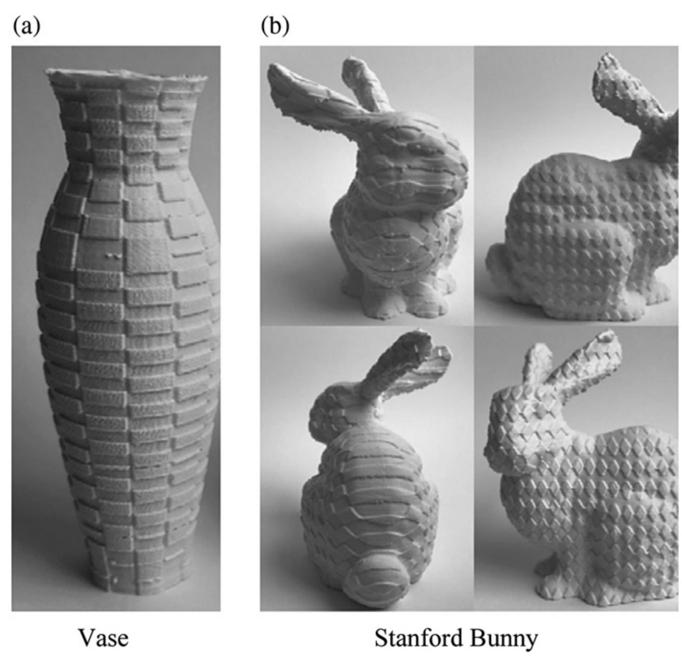

Figure 9. Fabricated objects using an FDM printer.

5. Conclusions

This article presented an easy-to-implement technique to create textured patterns 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 be easily 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). The method can also be adapted to other AM methodologies excluding 3-axis Directed

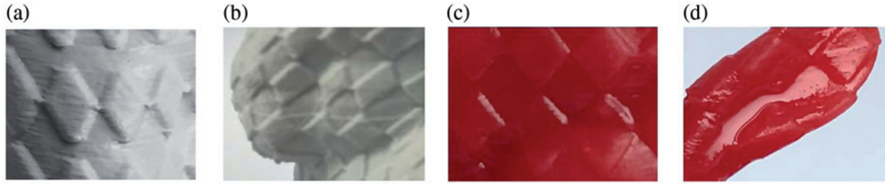

Figure 10. Some defects in FDM- and DLP-printed Stanford Bunny.

Energy Deposition (DED) since the fabrication of overhanging parts with steep angles is not possible with the state-of-the-art DED machines (Khanzadeh *et al.*, 2017).

As for future work, the method is to be a part of VEPRO/LIPRO command generation paradigm for production machinery (Yaman and Dolen, 2018) and is expected to be realized on small form-factor computers (like Raspberry Pi 3), which will drive any type of 3D printers such as DLP and FDM. The user will be able to switch easily from one technology to another by employing proper VEPRO/LIPRO (Python) libraries.

Funding

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

Notes on contributors

Ulas Yaman has been an assistant professor of mechanical engineering at the Middle East Technical University (METU), Turkey since October 2015. He received his Ph.D. (2014), MSc (2010), BSc (2007) in mechanical engineering, and minor in mechatronics (2007) degrees from METU. He was a visiting assistant professor of computer science at Purdue University from July 2014 to October 2015. His research interests include 3D printing and additive manufacturing, CAD/CAM architectures, command generation, FPGA-based embedded systems and computational geometry for design and manufacturing.

Melik Dolen was awarded his B.S. degree in mechanical engineering by Istanbul Technical University in 1990. He completed his M.S. studies in the University of New Hampshire (Durham) in 1994. He received his Ph.D. degree from University of Wisconsin - Madison in 2000. He is currently an associate professor in the Mechanical Engineering Department of the Middle East Technical University. His research interests include the design of advanced manufacturing systems, automation, precision engineering, and engineering optimization. He is the author or co-author of numerous technical papers in this field.

Christoph Hoffmann is a professor in the Department of Computer Science at Purdue University. Before joining the Purdue faculty, he taught at the University of Waterloo, Canada. He has also been a visiting professor at the Christian-Albrechts University in Kiel, West Germany (1980), and at Cornell University (1984–1986). His research focuses on geometric and solid modeling, its applications to manufacturing and science, and the simulation of physical systems. The research includes, in particular, research on geometric constraint solving and the semantics of generative, feature-based design. He is the author of *Group-Theoretic Algorithms and Graph Isomorphism*, published by Springer-Verlag and of *Geometric and Solid Modeling: An Introduction*, published by Morgan Kaufmann, Inc. He has received national media attention for his work simulating the 9/11 attacks on the Pentagon and the World Trade Center.

ORCID

Ulas Yaman http://orcid.org/0000-0001-8598-9281
Melik Dolen http://orcid.org/0000-0002-7413-3434

References

Angel, E. and Shreiner, D. (2014) *Interactive Computer Graphics with WebGL*, Addison-Wesley Professional, Boston, MA.

Armillotta, A. (2006) Assessment of surface quality on textured FDM prototypes. *Rapid Prototyping Journal*, **12**(1), 35–41.

Bermano, A.H., Funkhouser, T. and Rusinkiewicz, S. (2017) State of the art in methods and representations for fabrication-aware design. *Computer Graphics Forum*, **36**(2), 509–535.

Brunton, A., Arikan, C.A. and Urban, P. (2015) Pushing the limits of 3D color printing: Error diffusion with translucent materials. *ACM Transactions on Graphics*, **35**(1), 4.

Cheng, L., Wang, A. and Tsung, F. (2018) A prediction and compensation scheme for in-plane shape deviation of additive manufacturing with information on process parameters. *IISE Transactions*, **50**(5), 394–406.

Chia, T.L., Wang, K.B., Chen, L.R. and Chen, Z. (2003) A parallel algorithm for generating chain code of objects in binary images. *Information Sciences*, **149**(4), 219–234.

Gibson, I., Cheung, L.K., Chow, S.P., Cheung, W.L., Beh, S.L., Savalani, M. and Lee, S.H. (2006) The use of rapid prototyping to assist medical applications. *Rapid Prototyping Journal*, **12**(1), 53–58.

Hergel, J. and Lefebvre, S. (2014) Clean color: Improving multi-filament 3D prints. *Computer Graphics Forum*, **33**(2), 469–478.

Jee, H. and Sachs, E. (2000) Surface macro-texture design for rapid prototyping. *Rapid Prototyping Journal*, **6**(1), 50–60.

Jin, Y., Pierson, H.A. and Liao, H. (2017) Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders. *IISE Transactions*, **51**(2), 192–208. doi:10.1080/24725854

Kanada, Y. and Kanada, Y. (2016) 3D printing of generative art using the assembly and deformation of direction-specified parts. *Rapid Prototyping Journal*, **22**(4), 636–644.

Khanzadeh, M., Chowdhury, S., Tschopp, M.A., Doude, H.R., Marufuzzaman, M. and Bian, L. (2017) In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. *IISE Transactions*, 1–19. Advance online publication. doi: 10.1080/24725854.2017.1417656

Laput, G., Chen, X.A. and Harrison, C. (2015) 3D printed hair: Fused deposition modeling of soft strands, fibers, and bristles, in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 593–597, ACM, Charlotte, NC.

Livesu, M., Ellero, S., Martínez, J., Lefebvre, S. and Attene, M. (2017) From 3D models to 3D prints: An overview of the processing pipeline. *Computer Graphics Forum*, **36**(1), 537–564.

Mayer, A.G. (2016) U.S. Patent Application No. 14/992,200. https://patents.google.com/patent/US20130009338A1/en?q=Mayer&q=extrusion&q=rate&q=texture&oq=Mayer+extrusion+rate+texture

Ou, J., Dublon, G., Cheng, C.Y., Heibeck, F., Willis, K. and Ishii, H. (2016) Cilllia: 3D printed micro-pillar structures for surface texture, actuation and sensing, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5753–5764, ACM, San Jose, CA.

Panozzo, D., Diamanti, O., Paris, S., Tarini, M., Sorkine, E. and Sorkine-Hornung, O. (2015) Texture mapping real-world objects with hydrographics. *Computer Graphics Forum*, **34**(5), 65–75.

Pei, E. (2014) 4D Printing: Dawn of an emerging technology cycle. *Assembly Automation*, **34**(4), 310–314.

Reiner, T., Carr, N., Měch, R., Šťava, O., Dachsbacher, C. and Miller, G. (2014) Dual-color mixing for fused deposition modeling printers. *Computer Graphics Forum*, **33**(2), 479–486.

Schüller, C., Panozzo, D., Grundhöfer, A., Zimmer, H., Sorkine, E. and Sorkine-Hornung, O. (2016) Computational thermoforming. *ACM Transactions on Graphics*, **35**(4), 43.

Serra, J. (1982) *Image Analysis and Mathematical Morphology*, Academic Press, Orlando, FL.

Stanford Bunny, https://graphics.stanford.edu/data/3Dscanrep, accessed: 2018-04-20.

Takahashi, H. and Miyashita, H. (2017) Expressive fused deposition modeling by controlling extruder height and extrusion amount, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 5065–5074.

Van Herpt, O. (2015) Solid vibrations, http://oliviervanherpt.com/solid-vibrations, accessed: 2017-11-26.

Van Rompay, T.J., Finger, F., Saakes, D. and Fenko, A. (2017) "See me, feel me": Effects of 3D-printed surface patterns on beverage evaluation. *Food Quality and Preference*, **62**, 332–339.

Vase, https://www.turbosquid.com/3d-models/free-3ds-mode-kartell-misses-flower-power/1025377, accessed: 2018-04-20.

Vidimče, K., Wang, S.P., Ragan-Kelley, J. and Matusik, W. (2013) OpenFab: A programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics*, **32**(4), 136.

Yaman, U. and Dolen, M. (2016) A command generation approach for desktop fused filament fabrication 3D printers, in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pp. 4588–4593, IEEE, Florence, Italy.

Yaman, U. and Dolen, M. (2018) A novel command generation paradigm for production machine systems. *Robotics and Computer-Integrated Manufacturing*, **51**, 25–36.

Yaman, U., Dolen, M., Dilberoglu, U.M. and Gharehpapagh, B. (2017) A new method for generating image projections in DLP-type 3D printer systems. *Procedia Manufacturing*, **11**, 490–500.