

ORIGINAL RESEARCH ARTICLE

Advanced software development of 2D and 3D model visualization for TwinPrint, a dual-arm 3D bioprinting system for multi-material printing

Shaddin AlZaid¹, Noofa Hammad¹, Hamed I. Albalawi^{1,2}, Zainab N. Khan¹, Eter Othman^{1,2}, Charlotte A. E. Hauser^{1,2*}

¹Laboratory for Nanomedicine, Division of Biological and Environmental Science and Engineering, King Abdullah University for Science and Technology, Thuwal 23955-6900, Saudi Arabia

²Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

Abstract

This research highlights the development of a two-dimensional (2D) and three-dimensional (3D) preview software for additive manufacturing (AM). The presented software can produce a virtual representation of an actuator's path movements by reading and parsing the orders of the desired geometric code (G-code) file. It then simulates the coded sections into separate 2D layers and colored 3D objects in a graphical model. This allows users to validate the shapes before the 3D printing process. G-code is an operation language which is based on command lines of code written in an alphanumeric format. Each line of these commands controls one machining operation; this instructs the machine's motion to move in an arc, a circle, or a straight line to perform a specific shape after compiling all code lines. AM technology is widely used in most manufacturing fields (e.g., medical, chemical, and research laboratories) as a prototyping technology due to its ability to produce rapid prototyping models. 3D printing creates physical 3D models by extruding material layer by layer as 2D layers. At present, the most critical challenges in AM technology are drastically reducing prototyping materials' consumption and time spent. To address these challenges, the proposed software allows for visualization of G-code files and predicting the overall layers' shapes, allowing both structure prediction and subsequent printing error reduction.

***Corresponding author:**

Charlotte A. E. Hauser
(charlotte.hauser@kaust.edu.sa)

Citation: AlZaid S, Hammad N, Albalawi HI, *et al.*, 2022, Advanced software development of 2D and 3D model visualization for TwinPrint, a dual-arm 3D bioprinting system for multi-material printing. *Mater Sci Add Manuf*, 1(3):19.
<https://doi.org/10.18063/msam.v1i3.19>

Received: September 9, 2022

Accepted: September 21, 2022

Published Online: September 28, 2022

Copyright: © 2022 Author(s).

This is an Open Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: Whioce Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Keywords: Geometric code preview; Three-dimensional preview; Geometric code simulator; Three-dimensional printing

1. Introduction

Three-dimensional (3D) bioprinting is an additive manufacturing (AM) technology used in regenerative medicine and tissue engineering. This technique can overcome the limitations of traditional tissue replacement methods, autografts, and allografts as they often suffer from issues related to disease transmission as well as their scarcity^[1]. Aside from the heterogeneous nature of native tissues, they are made of various biomolecules, different cell types, and other biomaterials; this necessitates using multiple bioinks simultaneously or sequentially to 3D print biomimetic tissues^[2]. Compared with other

fabrication methods, 3D bioprinting methods that use multiple materials enable the fabrication of customized and complex tissues and organs, which facilitates the use of this technology in clinical and pharmaceutical applications^[3]. According to Liu *et al.*, the spatial locations and arrangement of biocomponents in the printed constructs are highly critical to achieve functional tissues^[4]. Customized computer-aided design (CAD) and 3D bioprinting give an additional advantage when used with cell-laden biomaterials, allowing for better control of the distribution of cells and biomolecules in 3D printed constructs^[3]. A comprehensive review paper by Kim *et al.* discusses the various 3D bioprinting techniques in general (i.e., using a single extruded material) as well as with multiple extruded materials^[3]. In addition, it shares the perspectives and future directions for developing state-of-the-art techniques of multi-material 3D bioprinting.

There are currently three broad categories, in which 3D printing techniques find themselves in: material extrusion, material jetting, and VAT polymerization^[5]. As of today, extrusion-based printing is the most commonly used printing technique. Its process of usage includes the loading of biomaterials into their respective cartilages and their subsequent extrusions from a nozzle using either mechanical or pressure-based forces. Since the nozzle is held above and moved along a predetermined platform, its motion allows the user to print a 3D object visualized by a viewing program on a connected computer. Some advantages of this technique include the ability to extrude bioinks laden with high cell density counts (hence their popularity in 3D cell culturing protocols), the capacity to lay those cells in specific areas within the overall 3D structure during its creation, as well as the relative cost efficiency of this technique as far as its customizability is concerned^[6].

Material jetting is another procedure in which bioprinted objects can be made, with inkjet techniques being widely used. Considered one of the earliest established bioprinting processes, this method includes a downstream droplet ejection induced by a volumetric change upstream of the nozzle^[7]. The formation of the droplet is most commonly generated through piezo-electric or thermally induced bubbles. For more in-depth information, refer to other papers^[8-10]. Some of the pros associated with this technique are attributed to its high printing speeds, low costs, and its contactless nature (thus avoiding contamination issues). Unfortunately, since the viscosity of the bioink cannot be above a certain value, inks laden with a high volume of cells cannot be utilized in the creation of the 3D object^[6]. As such, this method, as opposed to the aforementioned extrusion-based bioprinting procedure, is not optimal for printing biologically included structures.

The last in the umbrella of 3D printing techniques is dubbed VAT polymerization. Under it falls a number of categories including stereolithography (SLA) and digital light processing (DLP)^[5]. In both these technologies, a photo-initiated liquid material is solidified using a laser. To produce the overall 3D printed structure, the liquid is set against a platform which is raised after each layer of liquid is solidified by the laser^[11]. The difference between SLA and DLP, however, is that the former shines the laser at specific areas to produce the layers. In contrast, DLP instead shines the laser throughout the entire area and is only selective due to a mask between the laser and liquid. The downside associated with both these techniques, unfortunately, is the inability to include cells during the printing process^[11]. As such, other following protocols are needed to seed the cells in their intended locations.

One example of a bioprinter was illustrated by Liu *et al.* in which he designed a multi-material extrusion bioprinting platform that can accurately extrude and switch between several bioinks continuously and rapidly for fast complex tissue fabrication^[4]. Another 3D printing device is illustrated in the research study by Miri *et al.*, within which the development of a microfluidic device that can also rapidly switch between different hydrogel bioinks while maintaining high spatial resolution is elucidated^[12]. Interestingly, Ng *et al.* addressed the potential and advantages of adopting deep learning techniques into the 3D printing process; this was done in the intention of fabricating better biomimetic patient-specific tissues/organs^[13]. Similarly, An *et al.* highlighted the benefits of utilizing machine learning and digital twin of human tissues to mimic the 3D printing process and maximize the bioprinting efficiency and usage of biomaterials^[14].

In the standard 3D printing process, a geometric code (G-code) file (acquired from a slicer software) gives the directional map in terms of the desired 3D model's XYZ planes to the 3D printing software. In other words, the G-code file contains the geometrical description of a 3D model that allows the instruction of the 3D printer in its printing in a layer-by-layer fashion^[15]. If the G-code holds any syntax or semantic errors, the 3D model might not be correctively 3D printed^[16]. One way to detect G-code errors is by viewing and/or simulating its attributed 3D model based on its code; such a program is called a G-code visualizer/simulator^[16]. There are already plenty of 3D printing software programs with the features of viewing the 3D model and simulating the 3D printing process; one instance of this is the Repetier-Host Software. Another example of this was by Ueng *et al.*, in which a preview system for 3D printing was developed that utilizes G-code and some graphic techniques to efficiently display

the exterior and interior structure of a 3D model before printing^[16]. Similarly, in the research paper by Gahbiche *et al.*, script is built to implement the G-code commands into the finite element software; this is used to control the movements of the tool during the simulation of the sheet formation processes^[17].

Over the past years, we have developed an integrated 3D bioprinting system, consisting of a robotic arm and microfluidic pumps, and have been utilizing it to 3D print biological constructs. One application was using the system with a peptide-based material inside 3D printed support molds^[18,19]. Later, the system was extended to be a dual-arm printing system, called TwinPrint that controls two 3D bioprinting sets of printing arms. The motivation behind building the TwinPrint Software was to have a single software operating bioprinting devices of different structures. TwinPrint eases the tedious aspect of the printing process that deals with the usage of multiple software to control each device in a bioprinting set. For more information on TwinPrint, refer to this paper^[20]. Nonetheless, the TwinPrint Software lacks a 3D model previewer, which is crucial to check G-code errors and model corrections. In the CoraPrint method that we developed for coral restoration purposes, coral skeletons were 3D printed for the coral fragments to host^[21]. The asymmetrical shape of the coral adds to the complications 3D bioprinting provides, thus exemplifying this software's handiness in viewing the shapes. For this purpose, this paper proposes a 2D and 3D G-code preview software to be integrated within the TwinPrint System. The developed preview software displays the formation of the layers involved in developing the selected 3D model, layer by layer, and piles them up to form their collective 3D model.

The remainder of the paper is organized as follows: in section 2, the methodology shares how the G-code previewer software was built, starting from the procurement of the G-code to the development of the graphical user interface (GUI); in section 3, the preview software's results

(which are obtained from several G-codes' 3D models) are presented alongside in-depth analyses and discussions; and finally, section 4 provides a brief summary of the work and concludes the paper.

2. Methods

The visualization software was developed to preview 3D models before they were passed over to the computer numerical control machine. This allows for visualizing the G-code text file, which contains codified movement instructions for the tool path. However, unlike a simulator which shows the exact movement of the printer in the process of fabricating the intended object layer by layer, a viewer simply shows the path of the layers of which the objects are created (i.e., there is no movement shown, simply arrows). Nonetheless, these movement commands are essential for plotting 2D and 3D objects. Previewing the 3D model before printing is crucial as it is one way to check if the loaded G-code file is of the desired model. Moreover, errors in the G-code file can be detected when viewing the 3D model. Consequently, this increases the efficiency of printing by either reducing printing errors or avoiding printing of improper shapes, saving the user both time and materials.

2.1. G-code procurement

Initially, the 3D digital model images were obtained from scanning physical objects. A CAD software, SolidWorks, was then used to re-engineer the scanned models and export them as STL files. The STL files were then transferred to a slicing software (e.g., Slic3r) to slice the 3D virtual model and from there a G-code file was provided for the given model. The G-code file was then fed to the printing software to preview and print the 3D construct. The overall process needed to 3D print using a G-code is summarized in [Figure 1](#).

2.2. G-code parsing and coordinate calculations

A G-code parsing script was created to read the information inside the uploaded G-code file line by line. The script

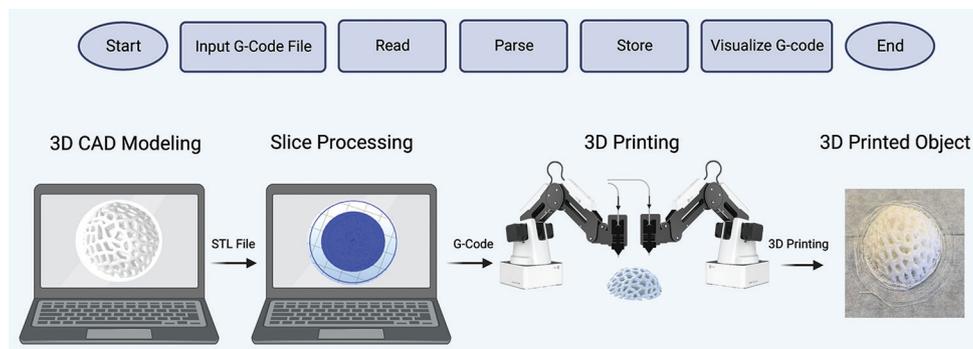


Figure 1. An overview of the three-dimensional printing process using the geometric code software.

extracts the information from the lines that contain Cartesian coordinates (X, Y, and Z) and stores this information in a file. In simple terms, if X, Y, and/or Z letters exist in a G-code line, the program extracts those lines alongside the G commands that specify the type of movement mentioned (e.g., linear or circular) and then determines whether positioning is absolute or relative. The positioning type determines how the Cartesian coordinates of the 3D construct are calculated. In case of absolute positioning, the XYZ values are specific coordinates and stored as they are. However, in relative positioning, the XYZ values of a line are added to corresponding XYZ of the previous line. Sometimes a G-code line might not have coordinate information for all XYZ; in this case, the values of the missing coordinates in a line are copied from the previous stored line information.

2.3. From G-code to visualization

The calculated XYZ coordinates and movement types were retrieved from either a.txt or.xlsx file to begin the translation process from written text lines into a 3D visualized model. The stored information represents the 3D trajectory points that form the path of the cross-section of the sliced surfaces. The sweep lines along the axes and generates the object's outer boundaries. The file is read sequentially, line by line, and an instantaneous conversion from a line to a visualized layer results in the high precision of the plotted shapes. The visualization was made using Scientific Graphics and GUI Library for Python (PyQtGraph) [22], and inspired by codes from these sources[23-25].

2.4. Software development

All the scripts and coding files used for the development of the 2D or 3D G-code preview software were written in Python Programming Language. This is also the same language used to build the TwinPrint System.

3. Results and discussion

The GUI of our program, as shown in Figure 2, has two main partitioned sections. The first section is used for previewing the 2D representation of the model and is shown singularly layer by layer through a white background and black lines, enabling a clear view; this can be seen in Figure 2A, 2C and 2E. The second section, likewise, shows a 3D preview by transferring the code line into path shapes as stated earlier, and shown in Figure 2B, 2D and 2F. The black lines, shown in Figure 2A and B, preview the shapes of the shell exteriors at those layers; they can be thought of as “the lines to be colored into” in the 3D viewing tab. Further detail of the 3D shape of the model (e.g., infills) can be seen as the “colored in” areas within the previewed 3D section (e.g., in Figure 2F). This illustration of detail represents the utility of a viewing program since shells

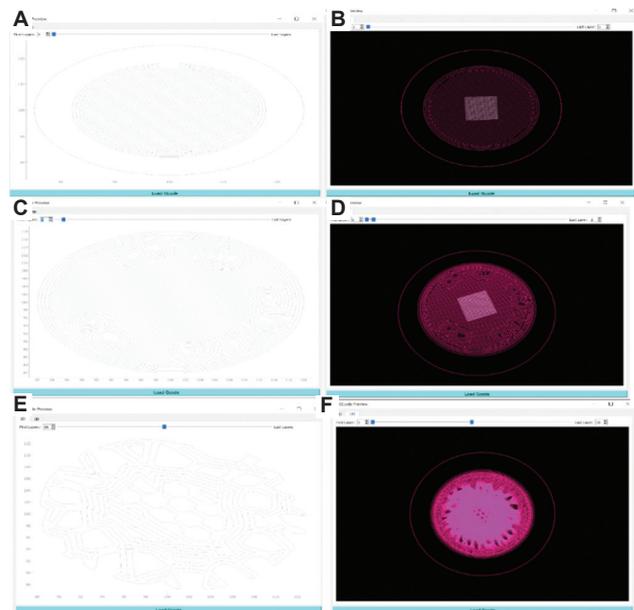


Figure 2. (A-F) Overview of graphical user interface for the two-dimensional and three-dimensional visualization sections.

remark on a model's rigidity, while infills take note of its stability, both factors of which are of high importance in 3D printing modules.

Another feature the software provides is an interactive interface with a dynamic graphics view. This interface allows the user to zoom in and out of the model, as well as to rotate around to observe different perspectives. The software also allows the isolation of specific layers to evaluate the initiating and finalizing paths (Figure 3A, 3C, and 3E) in addition to a scalable increase of layers for a path-centered view of the 3D object (Figure 3B, 3D and 3F). This comes with the ability to customize size, type, and color of the path lines. For ease of view, the user can also modify the background colors by selecting one of the two offered options. Axis labels can also be set for better navigation. Furthermore, the software is capable of visualizing 3D models with various structural complexities, ranging from very simple constructs (composed of straight lines and circles, e.g., Figure 2C) to irregular and curved structures (e.g., Figure 2D).

Another aspect accessible by plotting any given G-code's data is its capability of interactivity. In other words, such viewing measures allow the user to rotate and zoom the model and get closer looks at specific layers within it; such accessibility is shown in Figure 4.

Coupling the accessibility of viewing in the third dimension with the ability for line customization (e.g., size, type, and color) facilitates the preview of not only

single layers within the model but multiple as well. This feature is supplemented through the addition of intensifying color vibrancy as an indicator for increasing layer totals. In simpler terms, the more or brighter the colors, the more layers present in that model's area. This is illustrated in Figure 5, where colorful areas are denser with

extruded materials than their less colorful counterparts (Figure 5A and 5B).

To demonstrate the ease of viewing complex structures' layered compositions, Figure 6 was made. In Figure 6C, the particular sequence of layers (69 through 89) was chosen and refers to the highlighted G-code command lines in Figure 6A. The visualization program started with the command line (G1 = X86.980 Y110.590 E39.71886) for layer 69 and stopped at layer 89 using the command line (G1 = X108.649 Y112.360 E7.81012). Therefore, it plotted all command lines between these two layers. This feature allows users to see the specific details of any given layer. To elucidate, this allows the user to see the process of printing intermediate layers without the consideration of preceding or following layers. This can be practical since some of the objects have multiple features, such as in the Marmalade Reef, as shown in Figure 6F. This reef demonstrates the complexity a G-code can contain and illustrates the convenience of viewing the result of 3D printing using a single command line. For instance, while Figure 6B and 6C showcase the entirety and the fraction of the printed reef, respectively, a glance at either Figure 6A or 6D shows that the individual layers contained in the reef's overall shape, regardless of their intermediate nature.

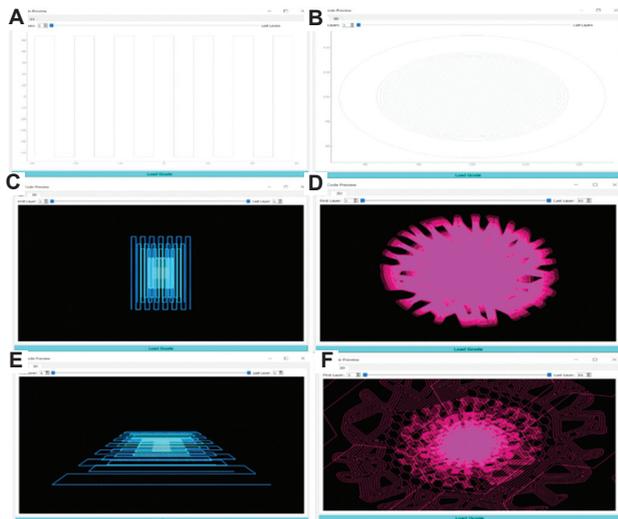


Figure 3. Visualization of single layers (A, C, and E) versus multiple layers (B, D, and F).

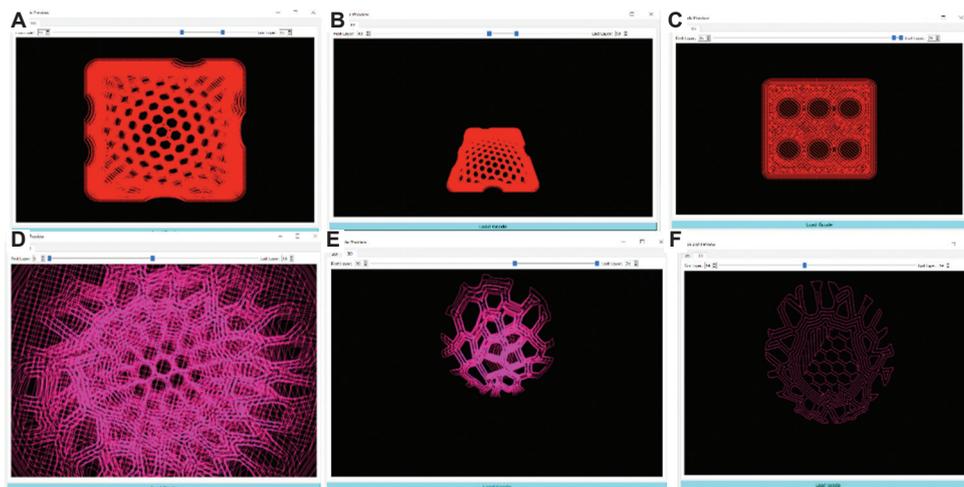


Figure 4. (A-F) Ability to zoom in and out using a three-dimensional preview.

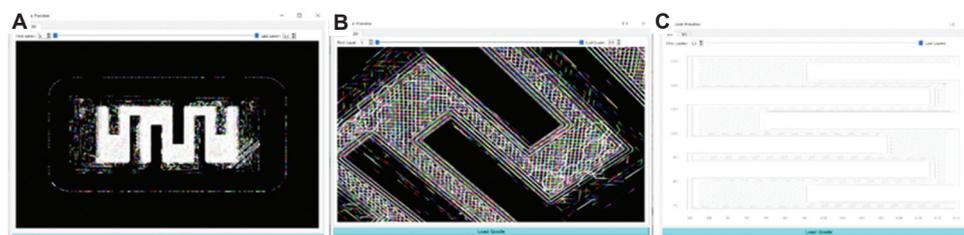


Figure 5. (A-C) Detailed view of multiple layers (three-dimensional) and single layers (two-dimensional) visualization.

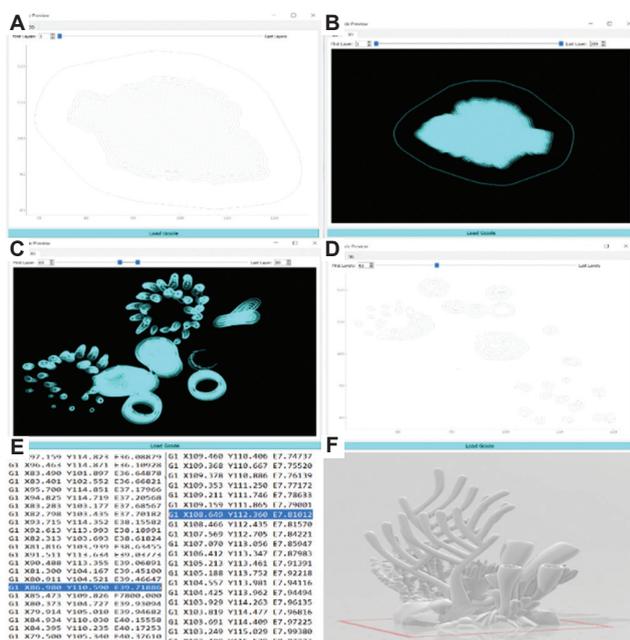


Figure 6. (A-F) Marmalade reef preview and command details for specific layers.

4. Conclusion

The proposed “2D and 3D Model Visualization Software” can visualize a G-code file by translating the commands into 2D and 3D objects before operating the machine. In the future, other features will be added, such as converting STL files into G-code, creating G-code shape paths, and modifying the current files of G-code. Adding this tool to the TwinPrint System reduces the use of several applications while printing, leading to a faster result that optimizes the printing to be as time efficient as possible while maintaining cost effectiveness. Moreover, the proposed method was optimized and improved to achieve the intended goal with the element of interactivity being added to the software.

Acknowledgments

The authors would like to acknowledge Ali Balubaid for his support.

Funding

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST).

Conflict of interest

There are no conflicts to declare.

Author contributions

Conceptualization: Zainab N. Khan, Shaddin AlZaid, Charlotte A. E. Hauser

Investigation: Shaddin AlZaid, Noofa Hammad, Zainab N. Khan

Methodology: Shaddin AlZaid, Hamed I. Albalawi

Software: Shaddin AlZaid

Visualization: Shaddin AlZaid, Hamed I. Albalawi

Writing – original draft: Shaddin AlZaid, Noofa Hammad, Hamed I. Albalawi

Writing – review & editing: Shaddin AlZaid, Noofa Hammad, Hamed I. Albalawi, Eter Othman, Charlotte A. E. Hauser.

References

- Kim Y, Nowzari H, Rich SK, 2013, Risk of prion disease transmission through bovine-derived bone substitutes: A systematic review. *Clin Implant Dent Relat Res*, 15: 645–653.
<https://doi.org/10.1111/j.1708-8208.2011.00407.x>
- Ashammakhi N, Kaarela O, 2018, Three-dimensional bioprinting can help bone. *The J Craniofac Surg*, 29: 9–11.
<https://doi.org/10.1097/SCS.0000000000004143>
- Tavafoghi M, Darabi MA, Mahmoodi M, *et al.*, 2021, Multimaterial bioprinting and combination of processing techniques towards the fabrication of biomimetic tissues and organs. *Biofabrication*, 13: 42002.
<https://doi.org/10.1088/1758-5090/ac0b9a>
- Liu W, Zhang YS, Heinrich MA, *et al.*, 2017, Bioprinting: Rapid continuous multimaterial extrusion bioprinting (Adv. Mater. 3/2017). *Adv Mater (Weinheim)*, 29:1604630.
<https://doi.org/10.1002/adma.201770016>
- Ng WL, Lee JM, Zhou M, *et al.*, 2020, Vat polymerization-based bioprinting process, materials, applications and regulatory challenges. *Biofabrication*, 12: 022001.
<https://doi.org/10.1088/1758-5090/ab6034>
- Jiang T, Munguia-Lopez JG, Flores-Torres S, *et al.*, 2019, Extrusion bioprinting of soft materials: An emerging technique for biological model fabrication. *Appl Phys Rev*, 6: 011310.
<https://doi.org/10.1063/1.5059393>
- Li X, Liu B, Pei B, *et al.*, 2020, Inkjet bioprinting of biomaterials. *Chem Rev*, 120: 10793–10833.
- Tekin E, Smith PJ, Schubert US, 2008, Inkjet printing as a deposition and patterning tool for polymers and inorganic particles. *Soft Matter*, 4: 703–713.
- Villar G, Graham AD, Bayley H, 2013, A tissue-like printed material. *Science*, 340: 48–52.
- Xu T, Jin J, Gregory C, *et al.*, 2005, Inkjet printing of viable mammalian cells. *Biomaterials*, 26: 93–99.
- Zhang LG, Leong K, Fisher JP, editors. 2022, 3D Bioprinting and Nanotechnology in Tissue Engineering and Regenerative Medicine. Academic Press, Cambridge, Massachusetts.

12. Miri AK, Nieto D, Iglesias L, *et al.*, 2018, Bioprinting: Microfluidics-enabled multimaterial maskless stereolithographic bioprinting (Adv. Mater. 27/2018). *Adv Mater (Weinheim)*, 30: 1870201–n/a.
<https://doi.org/10.1002/adma.201870201>
13. Ng WL, Chan A, Ong YS, *et al.*, 2020, Deep learning for fabrication and maturation of 3D bioprinted tissues and organs. *Virtual Phys Prototyp*, 15: 340–358.
<https://doi.org/10.1080/17452759.2020.1771741>
14. An J, Chua CK, Mironov V, 2021, Application of machine learning in 3D bioprinting: Focus on development of big data and digital twin. *Int J Bioprint*, 7: 342.
<https://doi.org/10.18063/ijb.v7i1.342>
15. Bikas H, Stavropoulos P, Chryssolouris G, 2015, Additive manufacturing methods and modelling approaches: A critical review. *Int J Adv Manuf Technol*, 83: 389–405.
<https://doi.org/10.1007/s00170-015-7576-2>
16. Ueng SK, Chen LK, Jen SY, 2017, A Preview System for 3D Printing. In: 2017 International Conference on Applied System Innovation (ICASI), p1508–1511.
17. Gahbiche MA, Boudhaouia S, Giraud E, 2020, A finite element simulation of the incremental sheet forming process: A new method for G-code implementation. *Int J Mater Prod Technol*, 61: 68–86.
18. Khan Z, Albalawi H, Valle-Perez A, 2022, From 3D printed molds to bioprinted scaffolds: A hybrid material extrusion and vat polymerization bioprinting approach for soft matter constructs. *Mater Sci Addit Manuf*, 1: 7.
<https://doi.org/10.18063/msam.v1i1.7>
19. Abdelrahman S, Alsanie WF, Khan ZN, 2022, A Parkinson's disease model composed of 3D bioprinted dopaminergic neurons within a biomimetic peptide scaffold. *Biofabrication*, 14: 044103.
<https://doi.org/10.1088/1758-5090/ac7eec>
20. Hammad N, AlZaid S, Alwazani H, 2022, TwinPrint: A Dual-Arm Robotic System for 3D Biofabrication, Viewed August 22, 2022. In: RobotoKAUST KAUST Research Conference on Robotics and Autonomy.
21. Albalawi H, Khan ZN, Valle-Pérez AU, *et al.*, 2021, Sustainable and eco-friendly coral restoration through 3D printing and fabrication. *ACS Sustain Chem Eng*, 9: 12634–12645.
<https://doi.org/10.1021/acssuschemeng.1c04148>
22. Lim J, 2022, Plotting with PyQtGraph Create custom plots in PyQt with PyQtGraph, 2020. Available from: <https://www.pythonguis.com/tutorials/plotting-pyqtgraph> [Last accessed on 2020 May 02].
23. Zhang Y, 2020, Gcode-Reader. Available from: <https://github.com/zhangyaqi1989/Gcode-Reader> [Last accessed on 2021 Feb 02].
24. Marvin Killing, 2014, QxtSpanSlider. Available from: <https://github.com/mkilling/QxtSpanSlider.py> [Last accessed on 2021 Sep 16].
25. Coolsunxu, 2021, Video_Cut. Available from: https://github.com/coolunxu/Video_Cut/blob/main/main.py [Last accessed on 2021 Sep 22].