3D Printing and Simulation of Naturally-Randomized Cellular-Automata

Abstract 3D-printing technology usually aims reproducing objects deterministically designed by 3D CAD tools; however, the author discovered that 3D printing can also generate self-organizing patterns similar to stochastic (or randomized) 1D cellular automata (CA). A method for generating patterns similar to randomized 1D or 2D CA by using a fused deposition modeling (FDM) 3D printer is thus proposed. With constant head motion and constant filament extrusion and without explicit randomness, this method generates very fine emergent patterns with natural fluctuation. By using this method, each time a different pattern is generated. In addition, a computational CA model that simulates the above process is also proposed. The proposed method will open a new horizon of 3D printing applications.

Keywords: 3D printing, self-organization, asynchronous cellular automata (CA), natural randomness, fluctuation, fused deposition modeling (FDM).

1 INTRODUCTION

3D printing technology, especially fused deposition modeling (FDM), was developed in 1980's and 1990's. Initially, An 3D printing method, or an additive manufacturing (AM) method, called "stereo lithography", was invented by Charles Hull in 1984 [Hul 86]. Since 1995, Stratasys Ltd. developed another 3D-printing method called FDM, and patented many techniques (e.g., [Cru 92]). In FDM, melted plastic filament is extruded by a hot nozzle to shape a 3D object. There are many other 3D-printing methods too.

FDM printers have become cheaper and are widely used now. Until recently, FDM technology was mostly monopolized by Stratasys Ltd. because of their patents. Although some of them are still alive, many other companies and open source communities recently developed cheap FDM printers. The most famous series of 3D printers are Makerbot [Mak] and RepRap [Rep].

3D printers can also generate fluctuated self-organizing patterns, but they are mostly ignored in 3D printing communities and industries. 3D printing technology usually aims reproducing objects deterministically designed by 3D CAD tools. However, printing processes may contain bifurcations, such as tearing filament, and printing conditions including nozzle temperature, extrusion process, air motion, and so on, are fluctuated. Although the printing process is usually controlled well so that the bifurcations are suppressed and the fluctuation usually does not cause serious problems, stochastic patterns caused by fluctuation can still often be seen in printed objects. The generated patterns are very interesting, but they have been always considered to be just noises and mostly ignored.

3D printing can be interpreted as cellular automata (CA), and fluctuated patterns are similar to patterns generated by randomized asynchronous CA. 3D printers generates zero-one spatial patterns: zero means no material exists and one means material, i.e., plastic, exists. The patterns may therefore be interpreted as CA [Wol 84]. Because 3D printers generates 3D patterns by accumulating 1D patterns, the simplest case can be interpreted as 1D asynchronous CA [Ing 84][Hof 87][Wik a]. The generated patterns are similar to some patterns generated by randomized asynchronous CA [Kan 94a][Kan 94b][Wik a].

In this paper, a method for printing seaweed-like stochastic patterns caused by the fluctuation is proposed. The proposed method can generate very fine emergent patterns similar to stochastic (or randomized) 1D or 2D cellular automata (CA) by using 3D printers with constant head motion and constant filament extrusion. This method generates natural fluctuation without explicit randomness. By using this method, each time 3D printers generate a different emergent and stochastic pattern.

The rest of this paper is organized as follows. Section 2 shows fluctuation in objects generated by 3D printing. Section 3 describes a method for generating fluctuated patterns using 1D and 2D CA. Section 4 shows experimental results using an FDM printer, and Section 5 describes a computational model that simulates the 1D (and 2D) CA generation. Section 6 concludes this paper.

2 Fluctuation Generated by 3D Printing

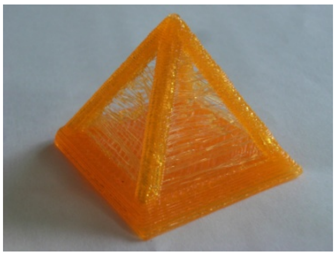
3D printing technology usually aims reproducing objects deterministically designed by using 3D CAD tools. Most of 3D CAD tools can generate a file in Standard Triangulated Language (STL) format, which is used by most 3D printing processes. STL approximates the surface of an object by a collection of triangles.

STL files are converted by a tool called "slicer" to G-code file, which represents a tool path (i.e., a path for the printer nozzle), and is used for input of most 3D printers. A G-code file contains a sequence of printer commands, which specify motion of the nozzle linearly from a point to another point with or without extrusion or specify print head temperature, cooling fan speed, and so on.

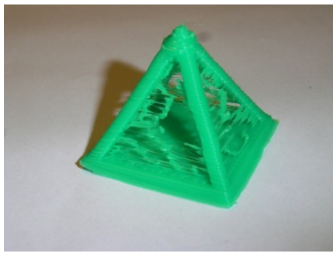
Although G-code file specifies a sequence of deterministic commands, the behavior of the 3D printer may be stochastic because printing conditions and process contains fluctuations and there are bifurcations that amplify the fluctuation. The fluctuation naturally causes stochastic patterns that can often be seen in printed objects.

Types of stochastic patterns are listed.

- **Patterns generated by excess extrusion**
  - *Strings*: A printing process includes head motions without extrusion. When the nozzle comes to a point in which extrusion must stop, the motion of plastic filament stops or is reversed. However, the extrusion process has delay. That means, filament continues to be extruded after the filament stops. This excess extrusion often causes thin strings, such as shown in the printed pyramids in **Fig. 1**(a) and (b). The string-generation process is stochastic because the condition of melted plastic fluctuates and the delay and the amount of excess extrusion are fluctuated. The filament material is polylactic acid (PLA) in Fig. 1(a), and it is acrylonitrile butadiene styrene (ABS) in Fig. 1(b). Both materials are commonly used for FDM.
  - *Chunks*: In contrast to strings, which are uniform, the nozzle may create small chunks of plastic. In Fig. 1(b), chunks are at the end or center of strings. Very small chunks are also formed in Fig. 1(a).
- **Patterns generated by lack of extrusion**
  - *Extrusion failure*: Sometimes filament does not extrude because of decreased temperature or any other causes.
  - *Sticking failure*: Sometimes extruded hot filament fails to stick to cooled filament. This prevents forming the designed shape.



(a) Printed using PLA



(b) Printed using ABS

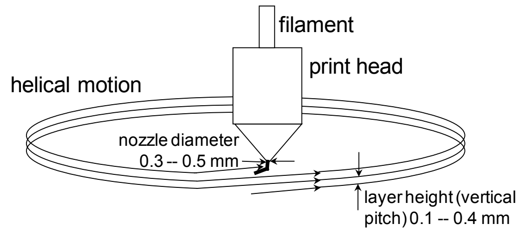
Fig. 1 Printed pyramids with strings and chunks

The generated patterns, especially those caused by excess extrusion, are very interesting because they suggest potential to generate more complex emergent patterns. However, they have been always considered to be just noises, and are mostly ignored.

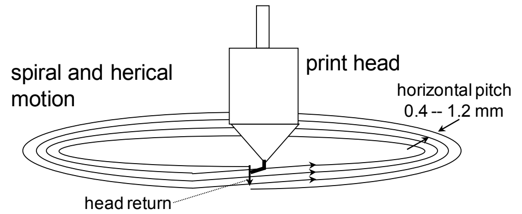
Although randomness may be explicitly introduced by algorithmic CAD tools, there are significant differences between this type of randomness and natural randomness caused by the printing process. Although most CAD tools are for human designers, there are generative or algorithmic design tools, such as OpenSCAD [Wik b], which can be used for generating randomized pattern by using random number generators. However, there are two differences; one is that, unlike explicit randomness, natural randomness is generated by a deterministically designed printing process similar to conventional method, and the other is that natural randomness generates much finer structures such as above patterns than explicit randomness.

3 1D/2D Pattern Generation Method

This section describes a method for generating fluctuated patterns using 1D and 2D processes. This method was designed to reduce artifacts including design itself as much as possible. To generate 1D CA patterns, a 1D space without edges is used ([Wol 84][Kan 94a] etc.). The space is topologically a circle. Therefore, to generate 1D pattern by an FDM printer, the print head draws circles in a clockwise or counterclockwise direction repeatedly and extrudes filament *constantly* (**Fig. 2**). The author discovered that, if the printing condition is selected carefully, the printer can generate self-organized stochastic patterns.



(a) 1D pattern generation method



(b) A modified method for 2D pattern generation

Fig. 2 1D/2D pattern generation method

The important condition or parameters for circular printing, which are constant, are as follows.

- *Nozzle diameter*: usually 0.3–0.5 mm
- *Filament cross-section*: much less than 0.2 mm^2 (0.5 mm by diameter)
- *Filament material*: ABS or PLA
- *Head temperature*: 220–260°C for ABS and 180–220°C for PLA
- *Layer height* (*h*): 0.1–0.4 mm

Head motion velocity is another parameter, but it is less sensitive unless printing process fails.

The initial state is all one; that means, the first layer of the circles is fully filled with plastic. The second and later layers are printed using the above parameters. These pseudo layers, which can be contrasted to strict layers used in conventional methods, are printed by a helical motion of the print head (Figure 2(a)). The reason why a helical motion is used is to avoid an artifact or a singular point, which is explained in Section 4.2.

The tool path, which is represented by a G-code file, is generated by a Python program. No STL file is used because print conditions including the tool path must be exactly designed. The tool path can be visualized by a G-code handling tool called Repetier Host. See **Fig. 3**. A Macintosh version of Repetier Host ver. 0.56 is used.

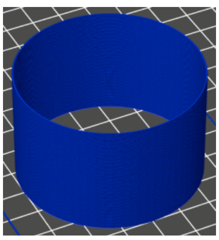


Fig. 3 Tool path used for the experiments

Because printed 1D plastics are fragile, a modified method that generates a more durable patterns is devised. Edge-less 2D patterns can be generated by using a torus, i.e., a two-dimensional edge-less shape, instead of circle in the case of 1D pattern. However, because it is very difficult to print a concentric torus by a 3D printer, instead, a spiral, which have edges, are used. 2D patterns can be generated, as shown in Fig. 2(b), by repeating to draw a horizontal spiral (instead of concentric circles to avoid a singular point effect) changing the nozzle height.

4 Experimental Results

1D and 2D experimental results are described and discussed.

4.1 1D printing results

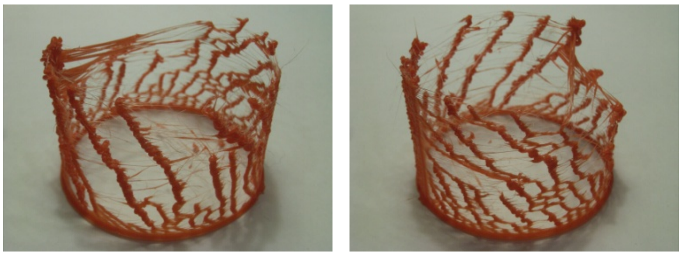
Experimental results of 1D printing were obtained by using the proposed method with the following conditions. Rostock MAX and Printrbot Plus 3D-printers were used, and the processes and products were demonstrated at Maker Faire Tokyo 2013 and a video was uploaded to YouTube [Das 13]. A circle was approximated by 72 (5°) straight line segments because a 3D printer nozzle cannot move along an arc. The parameters are:

- *Nozzle diameter*: 0.5 mm
- *Filament cross-section*: 0.1 mm^2
- *Filament material*: ABS or PLA
- *Head temperature*: 220–250°C.
- *Layer height* (*h*): 0.25–0.4 mm

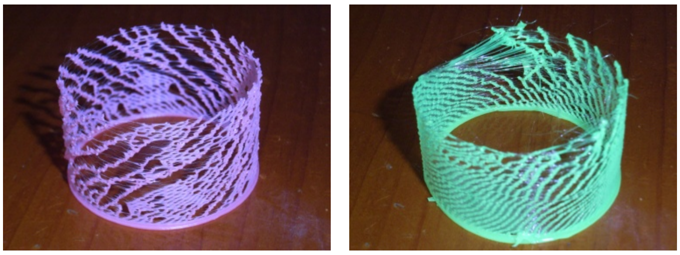
The temperature may be inexact because no exact measurement method was available. The motion velocity is 60–120 mm/s, which is several times faster than normal velocity. A high velocity is used for repeating trials quickly.

Four resulting patterns are shown in **Fig. 4**. The seaweed-like patterns flow clockwise when the printing direction is counterclockwise. When the printing direction is reversed, they flow counterclockwise. Seaweeds sometimes divide and sometimes merge. Therefore, mesh-like structure can be seen. The patterns may die in some layer; that is, the length of seaweeds are (may be) finite even if the printing process can be infinite. Because the amount of extrusion is independent from the number of seaweeds, excessive filament in upper layers forms thick strings. Patterns depend on the parameters, but they are not so much sensitive to the parameters as 1D synchronous or asynchronous CA.

Printing using PLA has also been tested (Fig. 4(b), right). It generates similar patterns, but they are closer to cyclic patterns, which seems to be less fluctuated.



(a) By counterclockwise motion (ABS, *h* = 0.4)



(b) By clockwise motion (left: ABS, right: PLA, *h* = 0.25)

Fig. 4 Generated pattern examples

4.2 Pattern destruction caused by singular points

Because the pattern flows when the head always moves in single direction, a steady pattern can be generated by bidirectional motion, i.e., by changing the direction quickly every time. **Fig. 5** shows a pattern broken at the right because of a singular point or an artifact; the nozzle turns there.

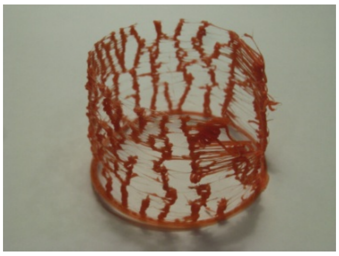


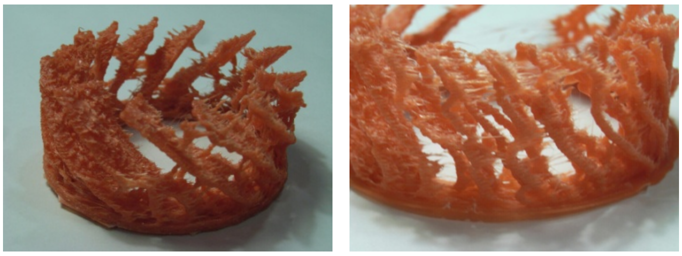
Fig. 5 Generated patterns by bidirectional motion

This destruction may be avoided by careful removal of singular points. However, it is quite difficult because there are many types of singular points, and they may cause significant effects on the generated pattern. For example, the head should move up gradually to avoid a singular effect caused by explicit vertical head motion (of 0.2–0.4 mm) caused by the layer transition. This means, to avoid singular points, the printing process should not follow the conventional layered 3D printing methodology.

Once the pattern is partially broken, it may be cured by the succeeding process or becomes completely broken depending on the printing conditions.

4.3 2D printing results

Experimental results of 2D printing were obtained using conditions similar to those of the 1D printing. Partial 2D patterns can be generated by printing concentric circles. **Fig. 6** shows the results. Fig. 6(a) shows a pattern that is more tightly-coupled along the diameter direction; the patterns on concentric circles are very similar. However, there are bridges between neighbors that break the similarity. Fig. 6(b) shows a pattern that is less tightly coupled; the patterns on concentric circles are different. However, it is difficult to reproduce this type of patterns. Tightly-coupled patterns are much easier to reproduce.



(a) Tightly-coupled pattern (b) Loosely-coupled pattern

Fig. 6 2D concentric circle pattern examples

5 Computational Model

This section describes a 1D CA model that simulates the printed patterns. In this model, the tool path is the same as that in the previous section. The circle is divided into and approximated by 288 (4 × 72) short line segments. In each segment, extrusion is on (one) or off (zero). The algorithm is described in **Fig. 7**. In this model, explicit randomization

```
extrudedFilament = 0;
for layer in 1, 2, ⋯, layers loop          // repeat for all layers
  z = 0.4 * layer;                          // layer pitch is 0.4
  for i in 0, 1, ⋯, 4 * 72 loop             // repeat for all parts of a circle
    cell[layer][i] = 0;                     // clear cells
    if extrudedFilament >= 1 then
                    // Amount of extruded filament is sufficient.
      if cell[layer–1][i] > 0 and random() <= p_current or
        // The cell at the same location of previous layer
        // is filled and the filament is successfully sticked.
        cell[layer–1][i+1] > 0 and random() <= p_next then
        // The cell at the next location of previous layer
        // is filled and the filament is successfully sticked.
          cell[layer][i] = 1;               // fill cell
          extrudedFilament = 0.0;           // clear filament
      end if
    end if
    drawNextArc(cell[layer][i], i, 2);      // draw circle part
    extrudedFilament = extrudedFilament + extrudedFilament1;
      // newly extruded filament (0 < extrudedFilament1 < 1)
  end loop
  cell[layer][steps] = cell[layer][0];
end loop
```

Fig. 7 3D printing pattern simulator (i.e., random number generator) is used instead of natural randomization.

Several examples of simulated results are shown in **Fig. 8**. In addition to the parameters introduced in the G-code for generating patterns, this simulation model introduced three more probabilistic parameters.

- $p_{current}$: Vertical grow parameter (= 0.97).
- p_{next}: Skewed grow parameter (= 0.6).
- extrudedFilament: Accumulated filament that has been extruded (= 0.15).

The values described above are those used in Fig. 7. Other values were also tested but splitting and merging patterns still could not be simulated. Tool paths were generated by Python program and visualized by Repetier Host.

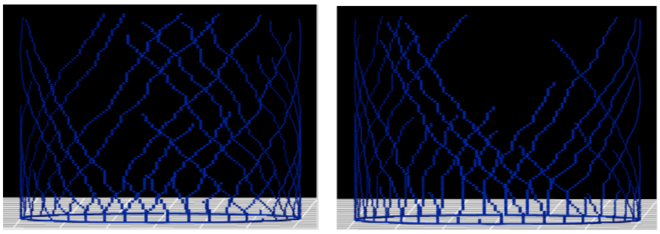


Fig. 8 Simulated patterns generated by 1D CA

The most important parameter is $p_{current}$. If its value is 1.0, the "seaweeds" almost vertically grow, as shown in **Fig. 9**. The average angle of seaweeds is simulated well when the value of $p_{current}$ is 0.97.

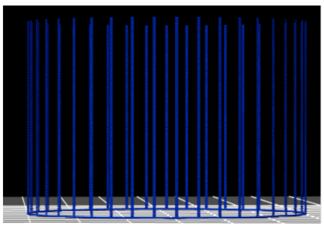


Fig. 9 Failed simulation result

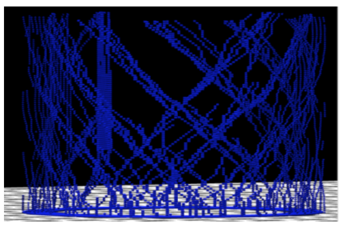


Fig. 10 Simulated 2D pattern

2D patterns can also be simulated by slightly modifying the program shown in Fig. 7. One more probabilistic parameter is introduced.

- p_{nextc}: Horizontal grow parameter, which means the probability of sticking filament to outer circle of the same position.

In contrast to the real printing results, it is difficult to simulate tightly-coupled pattern generation. Loosely coupled patterns are much easier to generate. Fig. 10 shows a result with p_{nextc} = 0.15 and extrudedFilament = 0.05. The vertical column at the backside is caused by a singular-point effect, i.e., layer connection points.

6 Conclusion

Even if the head motion and filament extrusion are constant, 3D printers can generate self-organizing and stochastic (or randomized) patterns. This paper proposes a method for printing seaweed-like fluctuated patterns using FDM, which are similar to those generated by 1D and 2D CA. This paper also proposes computational CA models that simulate the above process. 1D patterns can be simulated by the proposed model, but it still cannot simulate splitting or merging patterns. 2D patterns can also be partially simulated. In addition, a singular point that destroys patterns, such as an explicit head motion, was found.

Future work includes improvement of the computational model and developing applications. The proposed method will open a new horizon of 3D printing applications.

References

[Cru 92] Crump SS (1992) Apparatus and method for creating three-dimensional objects, US Patent 5121329.

[Das 13] Dasyn.com (2013) Creating naturally-fluctuated patterns using a 3D printer, YouTube, http://youtu.be/IJ15ysJR5l8

[Hof 87] Hofmann MI (1987) A Cellular Automaton Model Based on Cortical Physiology, *Complex Systems*, 1:187–202.

[Hul 86] Hull C (1986) Apparatus for production of three-dimensional objects by stereolithography, US Patent 4575330.

[Ing 84] Ingerson TE, Buvel RL (1984) Structure in Asynchronous Cellular Automata, *Physica D*, 10:59–68.

[Kan 94a] Kanada Y (1994) The Effects of Randomness in Asynchronous 1D Cellular Automata, *Artificial Life IV*.

[Kan 94b] Kanada Y (1994) The Effects of Randomness in Asynchronous 1D Cellular Automata, Extended version, http://www.kanadas.com/CA/AsyncCA/AsyncCAext.pdf

[Mak] http://www.makerbot.com/

[Rep] RepRap Wiki, http://reprap.org/

[Wik a] Asynchronous cellular automaton, Wikipedia, http://en.wikipedia.org/wiki/Asynchronous_cellular_automaton

[Wik b] OpenSCAD User Manual, WikiBooks, http://en.wikibooks.org/wiki/OpenSCAD_User_Manual

[Wol 84] Wolfram S (1984) Universality and Complexity in Cellular Automata, *Physica D*, 10:1–35.