

G-coordinator: G-code 生成の新たな手法とその可能性

谷口朝洋

December 28, 2023

G-coordinator: A New Method of G-code Generation and Its Possibility
Tomohiro TAGUCHI

Abstract

This study focuses on the development of a novel method for generating G-code, the program read by a 3D printer. We have developed an open-source software called G-coordinator, which utilizes Python and mathematical functions to create G-code directly and construct three-dimensional shapes. This alternative path generation technique, which is distinct from traditional slicing methods, offers the potential to tackle complex shapes that were previously challenging to produce and enables the formation of more precise printing conditions and paths. It also facilitates the straightforward realization of mathematical shapes.

要旨

本研究は、MEX 型 3D プリンタを用いて造形を行う際に、機械に読み込ませるプログラムである G-code の新たな生成方法に関するものである。Python と数学的な関数を用いて G-code そのものを直接作成し、3 次元形状を構築することできる G-coordinator (<https://github.com/tomohiron907/G-coordinator>) というオープンソースソフトウェアを開発した。従来の「スライス」とは異なる手法によるパス生成により、より精密な印刷条件の調整が可能になるとともに、いままでは難しかった形状や、数理的な形状の容易な実現が期待できる。

1 はじめに

1.1 G-coordinator とは

G-coordinator は、Python を用いて 3D プリンタ用 G-code を生成できるソフトウェアである。従来のスライスソフトによる G-code 生成と異なり、造形から印刷条件の調整まで 3D プリントに必要な工程の全てを G-coordinator ひとつで完結させることができる。Windows, Mac に対応したアプリケーションに加えて、Python 用のライブラリも公開している。Figure1 に G-coordinator (GUI アプリ) のスクリーンショットを示す。G-code を書き出す手順は主に 4 つのステップから構成される。

1. コードを書いて、造形を決定
2. グラフィックスビューで詳細を確認
3. 印刷条件の設定
4. G-code の出力

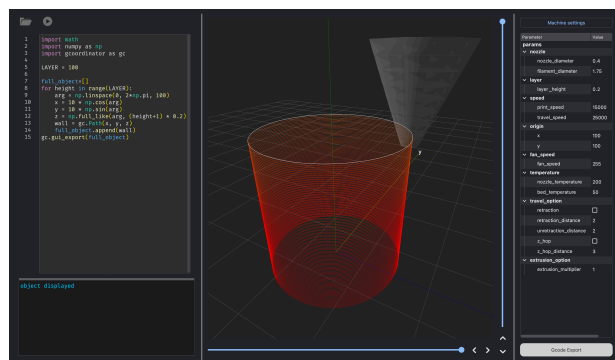


Figure 1: screenshot of G-coordinator

1.2 背景

一般的に、3D プリンタを用いる際には CAD 等で 3D データを作成し、その 3D データをスライスソフトで処理し、G-code を生成する。スライスソフトは、3D プリンタのノズルパスを自動で算出することができ、多くのユーザにとって、より簡単に G-code を作成することができる。しかし、より細かい制御やマニュアルでの制御を求めるユーザにとっては、スライスソフトでは 3D プリンタの挙動や印刷設定のマニュアル制御ができない。そこで、近年、3D モデルを介さず直接 G-code を作成する G-code Modeling[1] の手法が台頭している。G-coordinator は、G-code Modeling の新たな手法として位置付けられる。

2 目的

あなたのソフトウェアの目的や目標について述べます。

3 関連研究

あなたのソフトウェアと関連する研究について述べます。

4 提案手法

あなたのソフトウェアの概要やアルゴリズムについて述べます。

5 実装

あなたのソフトウェアの実装について述べます。

6 ユースケース

あなたのソフトウェアのユースケースについて述べます。

7 結論

あなたのソフトウェアの結論について述べます。

8 今後の課題

あなたのソフトウェアの今後の課題について述べます。

References

- [1] 知念司泰. ポリゴンモデリング技法をツールパス表現に直接変換するソフトウェアの開発, 2020.

9 付録

あなたのソフトウェアの付録について述べます。