

3DモデルへのGcode埋め込みによる一元管理とその利用例

高橋 治輝^{1,a)}

概要: 3Dプリンティングに利用される3Dモデルは形状に関する情報のみを表現しており、工作機械にはスライス処理によって得られるNCプログラム(Gcode)が送られる。いずれのデータ形式にも利点があるものの、これらを統一的に扱う方法はない。本稿では、熱溶解積層方式3Dプリンタとその制御言語であるGcodeを対象として、3DモデルデータにGcodeを埋め込む手法を提案する。埋め込みには3Dモデルのデータ形式のひとつであるAMFファイルのメタデータ領域を活用し、スライス処理時にこの情報を読み込んで使用する。メタデータへの埋め込み仕様について検討するとともにスライサのプロトタイプを実装し、Gcodeを直接埋め込む方法とスライス処理時にエフェクトを施す面の情報を埋め込む方法を紹介する。

1. はじめに

1.1 背景

3Dプリンタの普及によって、誰もが高精度な立体形状を作成できるようになっている。立体形状は頂点の集合と面を作る連結関係を記した3Dモデルファイルとして表現されており、モデリングシステムや3Dプリンタの入力データとして利用されている。3Dモデルは、スライス処理によって工作機械を動作させるためのNCプログラムに変換される。特に、Gcodeは工作機械を制御するための主流なプログラムであり、3Dプリンティングにおける熱溶解積層方式で採用されている。形状を表現する3Dモデルとは対比的に、NCプログラムは工作機械の動作を手続き的に表した命令群である。ひとつの命令には、次の移動先の座標や移動速度、加工方法に関する情報が含まれている。工作機械の創生期には、こうした低レイヤの情報も人手で記述されていたが、CAD/CAMシステムによる宣言的な記述の発展によって、人間がNCプログラムを用いて設計することは少なくなっていった [1]。

一方で、3Dプリンタに関するソフトウェア/ハードウェアのオープンソース化と一般ユーザへの普及によって、工作機械の動作をNCプログラムで直接制御することのメリットが明らかになってきた。たとえば、Gcodeのレベルで3Dプリンタを制御することで、毛のように細い構造の造形が可能である [2]。また、3Dプリンタを作品制作に活用するアーティストは、Gcodeによる直接制御を駆使してさまざまな表現を試みている [3][4]。田中らは、Gcodeの直接記述を可能とするプラグイン「fabrix」を提案すると

ともに、「現在の熱溶解積層方式の工程は、機械自体の可能性を制限している」と指摘した [5]。

1.2 3DモデルとGcodeの統合

3Dプリンタの場合、スライサのパラメータを調整によって狙い通りのGcodeを作るテクニックが存在する。たとえば、ノズル口径と同じ太さの細長い3Dモデルを処理すれば、1本の線を造形するGcodeが得られる。あるいは、意図的に上面と内部を造形しないように設定すれば、コップのような形状が得られる。Gcodeレベルでの制御を用いた造形手法が発展するとともに、3Dモデルと単純な変換だけでは工作機械の能力を十分に発揮できないことも明らかになってきた。むしろ、Gcodeで直接記述したほうが容易であったり、高度な造形が行える場合がある。さらに、3Dモデル自体がどのようにGcodeへと変換されるべきか、という情報を持っていれば、その情報をスライサで読み込んで処理することができるだろう。しかし、現在の造形工程において、造形する形状と実際の工作機械の動作には、3DモデルとNCプログラムというデータ形式の隔たりがあり、これらを統一的に扱える方法やファイル形式はない。

本稿では、熱溶解積層方式3Dプリンタを想定し、3Dモデルファイルの中にGcodeを埋め込む手法を提案する。3Dモデルを表現するデータのひとつであるAMFファイルのメタデータの領域に、Gcodeやスライス処理時に活用できる情報を記述する。埋め込まれた情報は、スライスの際に読み込まれて最終的なGcodeとして統合される。以降、Gcodeを埋め込むためのファイルの仕様についてのアイデアとユースケース、提案手法を実現するためのプロトタイプについて述べる。

¹ 明治大学

^{a)} haruki@meiji.ac.jp

2. 関連研究

本稿の関連研究として、3D モデルのファイル形式、Gcode を用いた造形に関する研究についてまとめる。

2.1 3D プリンタで利用される 3D モデルのファイル形式

3D プリンタのためのデータには伝統的に STL^{*1}形式が使われてきた。1980 年代に 3D Systems によって開発されたフォーマットで、三角形ポリゴンをまとめて立体形状を表現する。面を単位として法線や頂点を管理しているため容易に扱うことができるが、色や材料、製作者の情報など、現代の 3D プリンティングに必要な情報を持つことができない。こうした課題を解決するべく、3D プリンタのための新しいファイル形式が模索された。STL 2.0[6] をはじめとする議論の末、2011 年に AMF (Additive manufacturing file format)^{*2}が開発される。AMF は XML で記述されており、STL が持っているの幾何情報を完全に表現できるほか、3D モデルに関するさまざまな追加情報を管理できる。また、拡張性や互換性に優れていることも特徴として挙げられる。さらに 2016 年には、富士ゼロックスにより新たなデータ形式 FAV[8] が開発された。ボクセルを基本として 3D データを表現するファイル形式であり、データを高精度に表現する可能性を担保している。

本稿は、こうした 3D プリントに用いるファイル形式に関する研究として位置付けられる。また、多くのソフトウェアで対応が進む AMF ファイルをベースとして、Gcode を埋め込む手法を構築する。

2.2 Gcode を活用した造形

熱溶解積層方式 3D プリンタは Gcode という NC プログラムを用いて造形を行う。このデータは、3D プリンタのノズルの移動先に相当する座標や工作機器の設定に関する命令を手続き的にまとめた形式である。3D モデルデータに依存しない動作が可能であり、これを活用したさまざまな造形手法が提案されている。たとえば、ノズルの先端で樹脂を引きずることで髪の毛のように細い構造を作り出す手法 [2][9] や、ノズルの高さや樹脂量制御によって作られる構造を用いた手法 [10] がある。これらは、ノズルの動きや樹脂の押し出し具合を Gcode を介して直接的に制御することで実現されている。また、3次元空間中の移動を駆使してワイヤフレーム造形を実現する手法 [11] では、造形の高速化のために Gcode 制御が用いられた。さらに、2色の樹脂を用いて波打った壁面を造形することで、色調表現 [12] が可能となるように、単純な積層造形にとどまらない動作をさせることも可能である。すでに造形されたもの

や日用品を拡張する造形手法である Encore[13] や、紙で組んだ電子回路を挿入してインタラクティブな造形物を作り出す 3D Folded PrintGami[14] では、Gcode の編集で一時停止命令を挿入することで、新しい造形工程を可能としている。こうした造形は、Gcode のような工作機械を直接制御する言語の記述で実現されており、3D モデルなどのデータとして表現することが困難であるという特徴がある。

造形手法を駆使した表現は、アーティストの作品にも見ることができる。LIA[3] は、“Filament sculptures” と題した作品群を制作した。制作過程はブログ記事^{*3}にまとめられており、いずれも Gcode レベルでの制御によって実現されている。O’Dowd は造形手法を駆使した Textured 3D printing[4] について研究を行っている。造形された材料は機械の動きの物理的な痕跡であると述べ、この動きを直接制御することで美的なテクスチャ表現を可能とした。

データの性質上、Gcode を直接編集することは難しい。そのため、Gcode の直接記述を可能とするシステムが提案されている。田中らは、Adobe Illustrator で作られた曲線を Gcode へと変換変換するプラグイン「fabrix」を発表した [5]。金田は、プロシージャルな 3D プリントを実現するための Python ライブラリを提案している [1]。Grasshopper のプラグイン「Silkworm」は、CAD ソフトウェアで記述された曲線やジオメトリを Gcode に変換できる [15]。著者は、Gcode に用いられるパラメータをシンセサイザの要領で探索可能とするシステムを提案した [16]。

3. AMF への Gcode の埋め込み

AMF ファイルへ Gcode を埋め込むために、このファイルの仕様とデータの表現方法についてまとめる。

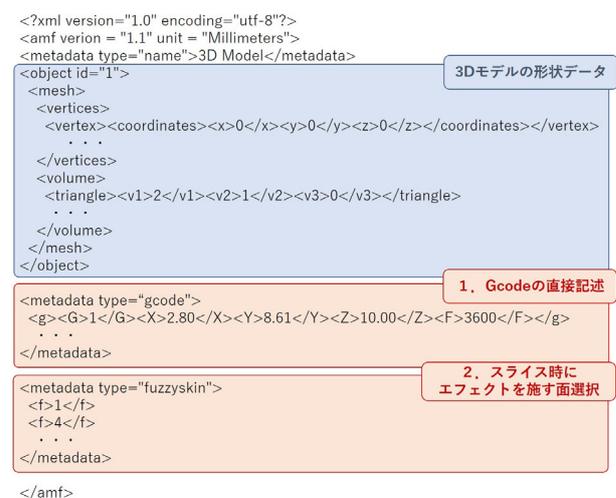


図 1 AMF ファイルの基本的なデータ形式とファイル内に埋め込んだ Gcode に関する情報 (赤枠)。

^{*1} [https://en.wikipedia.org/wiki/STL_\(file_format\)](https://en.wikipedia.org/wiki/STL_(file_format))

^{*2} https://en.wikipedia.org/wiki/Additive_manufacturing_file_format

^{*3} <http://liasomething.tumblr.com/>

3.1 AMF ファイルの仕様

AMF のファイル構造を図 1 に示す。ファイル全体は XML 形式で記述されており、3D モデルデータを表現する方法は STL と同様に頂点と面を基本としたポリゴンメッシュである。<object> 要素の子要素 <mesh> 要素には、頂点と面に関する情報が <vertices> 要素と <volume> 要素として格納されている。<vertices> 要素には各頂点が座標軸ごとに記述されており、<volume> 要素の子要素 <triangle> 要素で、面を構成する頂点の番号が記述される。さらに、<metadata> 要素を用いることで、AMF ファイルに任意の属性のメタデータを記述することができる。この要素は、3D モデルデータの名称や作成者、権利情報など記すために使用される。

3.2 埋め込みフォーマット

本稿で提案する手法は、AMF ファイルの <metadata> 要素を活用して、Gcode に関する情報を 3D モデルデータの一部として埋め込む。ここでは、実験的に 2 つの記述方法を試みる (図 1 の赤枠)。

ひとつは、文字通りに Gcode を直接記述する方法である。Gcode は、パラメータを意味する英字と数値がスペース区切りになった形式であるため、これを XML 形式に書き直して以下のようにメタデータ領域に記述する。

```
<metadata type="gcode">
  <group>
    <g><G>1</G><E>2</E></g>
  </group>
</metadata>
```

<metadata> 要素の属性は便宜的に *type*="gcode" とした。ひとつの命令は、<g> 要素内にパラメータ名とその値の組み合わせで記述する。上記の例は、Gcode での *G1 E2* に相当し、樹脂を 2 mm だけ押し出す命令として解釈される。同様に、Gcode で用いられるパラメータを記述することが可能であり、移動先の座標データ (X, Y, Z) や移動速度 (F)、その他マシンの設定を行うための命令も同一のフォーマットで記述できる。また、<group> 要素は、複数の命令をまとめて処理することを想定したグルーピングのために利用する。スライサは、*type*="gcode" 属性の <metadata> 要素を見つけて、記述された <group> 要素を順次読み込んでいくことで、直接記述された Gcode を処理する仕組みとなる。

もうひとつの埋め込み方法は、3D モデルを Gcode へと変換するスライサ処理を想定して、この処理に利用可能な情報を 3D モデルの中に明示しておくものである。たとえば、造形物の表面のテクスチャ表現を、Gcode レベルで行う造形手法がある [10][17]。こういったテクスチャ表現手

法の多くは 3D モデル全体に処理が施される。そこで、3D モデルの面の単位でエフェクトを掛けるかどうかを決めておき、この情報を以下のように格納しておく。

```
<metadata type="fuzzyskin">
  <f>1</f>
</metadata>
```

<metadata> 要素の属性には、スライス時に施すエフェクト名を指定するような仕様とした (Fuzzy Skin については次章で述べる)。ひとつの面は、<f> 要素内に番号を記述することで埋め込む。この番号は、<volume> 要素の中に記述された三角形の順番に対応している。この番号さえわかれば、対応する面およびその面を形成する 3 つの頂点を参照することができる。

埋め込まれた情報を活用するためには、専用のスライサを用意する必要がある。しかし、このままの形式であっても従来の CAD ソフトウェアやスライサで通常の AMF ファイルとして読み込むことが可能である。この場合、メタデータに埋め込まれた情報は無視される。

4. ユースケース

ここでは、前述の仕様に基づいて Gcode を埋め込んだ 3D モデルファイルを活用する例を示す。具体例として、直接記述された Gcode を読み込んで追加する手法と、3D モデル内で選択された面に対してのみエフェクトを掛ける手法について述べる。各 3D モデルデータの作成には、CAD ソフトウェアの Rhinoceros^{*4}とそのプラグインである Grasshopper を使用した。スライス処理には、専用のスライサを実装して使用した (次章で詳細に述べる)。3D プリンタは熱溶解積層方式の Creality CR-10S、ホストアプリケーションは Repetier-host^{*5}、樹脂は商用の PLA 樹脂を使用した。

4.1 Gcode での花卉デザイン

図 2 は、熱溶解積層方式 3D プリンタを用いて花の造形を行った例である。このデータには、花の中心部分が 3D

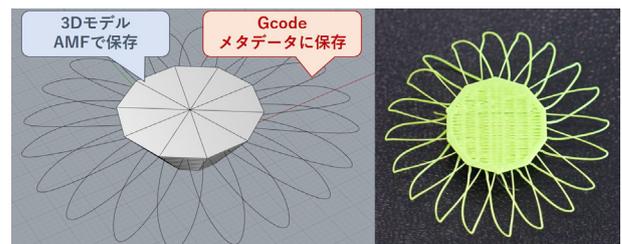


図 2 3D モデルに Gcode を書き加えた花の例。

*4 <https://www.rhino3d.co.jp/>

*5 <https://www.repetier.com/>

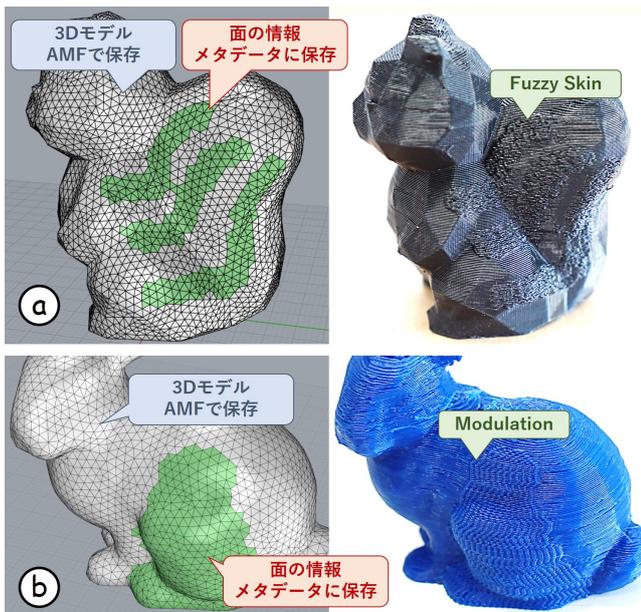


図 3 面の情報と Gcode 制御を用いたエフェクト。(a) Fuzzy Skin エフェクトを施した様子。(b) Modulation をかけて周期的な模様を付加した様子。

モデル、花卉の部分が Gcode としてまとめられている。中心部分から外側の空中に向かって造形を行い、再び中心部分へと引き返す動作が Gcode によって記述されており、この動きが花卉を作り出す。スライサは 3D モデル部分を Gcode へと変換するとともに、埋め込まれた Gcode を調べていく。それぞれの花卉に対応する Gcode は、メタデータ内の `<group>` 要素にまとめられており、Gcode を追加すべき高さに到達したら、中心部分と統合して出力する。なお、3D モデル部分は Rhinoceros のメッシュツール、花卉の部分は B スプライン曲線を用いて作成しているため、3D モデルと Gcode の編集が同一ソフトウェア上で行われていることも特徴としてあげられる。

4.2 Gcode 制御を用いたテクスチャ表現

3D モデルにスライス処理を想定した情報を埋め込む例として、Gcode 制御を用いた 2 つのテクスチャ表現について紹介する。

4.2.1 Fuzzy Skin

Fuzzy Skin は、Ultimaker Cura[17] というスライサに実験的に搭載されているテクスチャの表現方法であり、造形物の外壁を作る際の移動に対して、X、Y 軸方向への適当なノイズを乗せる。このノイズによって、3D プリンタのノズルは細かく振動するようなパスを描き、造形物の表面はざらついた仕上がりとなる。Cura におけるこのエフェクトは、入力された 3D モデル全体に施されるが、どの面に対してエフェクトを施すかという情報を持ってれば、その部分のみエフェクトを施すことができる。図 3a は、リスの 3D モデルの面を選択して、その情報を 3D モデル

に埋め込んだ例である。スライサはこの部分のみを別途処理し、Gcode 制御で模様を作り出す。

4.2.2 Modulation

Fuzzy Skin と同様な仕組みで、造形物の外壁を作る際の移動に対して周期的なパターンで変調を施す。先行研究 [12] では、このエフェクトをテクスチャの情報を用いて行っており、さらに 2 色の樹脂を用いることで色調の制御を実現していた。ここで紹介するエフェクトは、色調やテクスチャの表現には至っていないが、周期的なパターンによる均質なざらつきを持った表面を、指定した面にのみ作ることができる。図 3b は、Stanford Bunny の 3D モデルの足部にのみ Modulation エフェクトを施した例である。

5. 実装

Gcode を埋め込んだ AMF ファイルの生成、およびこれを用いた造形を検証するため、Rhinoceros と Grasshopper を使用してプロトタイプを実装した。Grasshopper は、パラメトリックな 3D モデリングを可能とするビジュアルプログラミング言語であり、さまざまな処理がコンポーネントとして用意されている (図 4)。また、CAD システムの Rhinoceros と連携されており、モデリングしたデータを受け取ることでより複雑な処理を行うことができる。こうした機能を利用するとともに、C#を用いてコンポーネント自体を記述することでシステムを実装する。プロトタイプシステムは、AMF ファイルに Gcode を埋め込んで書き出しを行うプログラム、埋め込まれた Gcode を考慮して 3D モデルを処理するスライサからなる。

5.1 AMF ファイルへの埋め込みと書き出し

AMF ファイルの作成のために、Grasshopper に 3D モデルデータと埋め込む対象となるデータを入力する。Grasshopper に入力された 3D モデルのデータは、メッシュを分解するコンポーネント (Deconstruct Mesh) を通して、頂点、面の情報、法線に分解される。分解されたデータは、C#で実装した AMF 書き出しコンポーネントに入力され、XML 形式に整形された後にファイルとして保存される。

埋め込む Gcode のもととなるデータは、Rhinoceros 上で記述された曲線 (Curve) を利用する。Grasshopper に入力された曲線のデータは、一定間隔に分割されて頂点のリストに変換される。この曲線を始点から順に Gcode の形式に変換し、XML 形式に整形した後に AMF 書き出し

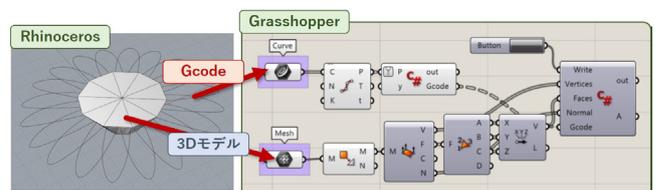


図 4 AMF ファイルを書き出す Grasshopper のプログラム

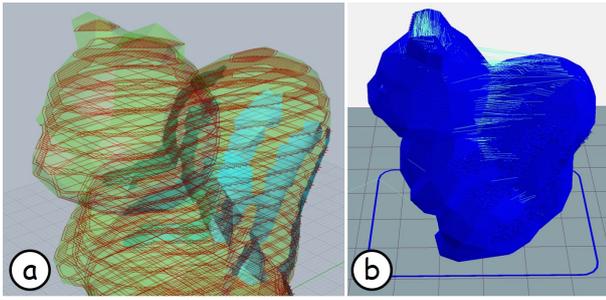


図 5 スライスと造形プレビュー。(a) Rhinoceros 上で 3D モデルと Gcode を表示した様子 (可視化のために粗い積層ピッチで処理を行っている)。(b) Repetier-host での造形プレビュー。

コンポーネントに入力することでメタデータ領域に書き込む。図 4 には、この処理を行う Grasshopper のプログラムを抽出して示した。なお、Gcode への変換処理に関して、現在は一つの造形パスが約 1 mm になるように分割しており、0.4 mm ノズルかつ積層ピッチ 0.3 mm でスライスした状態に等しくなるように樹脂量を設定している。また、移動速度は 1200 mm/min、造形速度は 500 mm/min と設定した。これらは実験的に決定した値であり、造形する対象や樹脂によって調整が必要となる。

入力された 3D モデルの面に特定の色がついている場合は、その面をスライス時に利用可能な追加情報として埋め込む。面の着色には、Rhinoceros のプラグインである MeshPaint3D^{*6}を使用した。着色された面をメッシュの中から取り出し、対応する番号を XML 形式に整形して AMF ファイルのメタデータとして書き込む。なお、現在はスライスに実装されている Gcode エフェクトが既知であることを想定している。そのため、属性として設定するエフェクト名は Grasshopper 上に用意したリストから選択する。

5.2 スライサの実装

前述のシステムを用いて書き出された AMF ファイルから Gcode を作り出すためには専用のスライサが必要となる。そこで、Rhinoceros と Grasshopper を使用して汎用的に使用可能なスライサを実装し、その機能のひとつとして AMF ファイルに埋め込まれた情報を読み込んで処理する機能を搭載する。図 5 は、スライサの処理の様子と造形に利用されるパスをプレビューした様子である。現在の実装では、サポート材の付加や造形速度の調整などの最適化を行っていないが、図 2 や図 3 で示したように、造形が十分に行える程度の Gcode を生成することができている。

5.2.1 スライサの基本部分

AMF ファイルの書き出しと同様に、スライサも Grasshopper に標準搭載されたコンポーネントの組み合わせを中心に実装した。まず、入力された 3D モデルファイルからメッシュデータを読み込み、任意の積層ピッチで作出した平

面群との交点を取る (Mesh | Plane Intersect)。この交点を、各平面上でつなぎ合わせていくと 1 層分の輪郭が得られ、これが Gcode 中の外壁のデータに相当する。また、この輪郭をノズルの口径分だけ内側に押し込み (Polyline Offset)、外壁に 2 周分の厚みをもたせた。

造形物の底面や上面、さらに上下の層とずれている部分は塗りつぶすように造形し、内部構造も適度な空間を作りながら全体を埋めるように造形する必要がある。このために、各層の高さに対応する平面を埋め尽くすように十分に長い線分を生成する。この線分と輪郭との交点を計算 (Curve | Line Intersect) すると、輪郭によって線分が切り取られる。これらを交互につないでいくことで、1 層を埋め尽くすようなパスが作られる。

外壁と内部のパスを作成した後、C# で実装したコンポーネントに入力して Gcode を生成する。一般的なスライサと同様に、外壁と内部のパスを順に繋ぎ、次のパスへの移動の際には樹脂溢れを防ぐための引き戻し命令を挿入する。Stanford bunny を始めとする 3D モデルを用いてスライス処理を行ったところ、いずれの 3D モデルも正しく、現実的な処理時間で Gcode が生成できることを確認した (50 mm 四方の大きさの Stanford bunny を 0.2 mm ピッチで処理したとしても、数十秒あれば処理が完了する)。

5.2.2 埋め込まれた情報の利用

前述の処理で、造形物の内部・外部の Gcode が作られる。スライサは、さらに AMF ファイルに埋め込まれた情報を参照して Gcode に修正を加える。まず、埋め込まれた情報を幾何情報と同様に AMF ファイルからパースし、`<metadata>` 要素の属性に応じて処理を行う。

埋め込まれた情報が Gcode を直接記述したデータであった場合、スライサは `<group>` 要素ごとに Gcode をまとめあげて、3D モデルの処理で得られた Gcode の中に追加する。現在は、造形物との接触問題を考慮し、記述された Gcode の始点の高さを基準にして Gcode を結合している。

埋め込まれた情報が選択された面とエフェクトに関するデータであった場合、スライサは 3D モデルの処理と並行してエフェクトを加えた Gcode を生成する。この Gcode は、前述の基本部分と同様の Gcode 生成を、選択された面のみを使用したメッシュの部分集合に対して行うことで作られる。なお、エフェクトは表面にのみかかるため内部の処理は行わない。得られた輪郭を短い区間で分割し、各頂点を X,Y 座標に対してずらすなどの処理を施すことで、エフェクトが施された Gcode を得る。いずれも同じ 3D モデルのメッシュから作られているため、3D モデルから得られた Gcode とエフェクトが施された Gcode は一部重複する。この重複が生じる部分を調べ、差し替えることで 2 つの Gcode を結合する。

*6 <https://www.food4rhino.com/app/meshpaint3d>

6. 展望と今後の課題

本稿で提案した仕様およびシステムを用いて、3DモデルにGcodeを埋め込むことが可能となった。メタデータへ情報を組み込むためのシステムやそれを適切に読み込んで処理を行うスライサが必要であるが、メタデータの領域を活用していることから、従来のAMFファイル自体としても利用できる形で3Dモデルデータの拡張が行えている。

3Dモデルを代表としたCADデータは、デジタルファブリケーション文化を支える重要な要素であり、個人レベルでの設計やWeb上でのやり取りでも盛んに利用されている。一方で、さまざまな工作機械の登場、その扱い方が模索されていく中で、形状のみを表現する3Dモデルと単純な変換処理だけでは、工作機械の能力を十分に発揮できないことも明らかになってきた。共有のしやすさやパフォーマンスが優れたデータとしてだけでなく、製品仕様や図面の情報を3Dモデルとして管理するModel-based definition^{*7}の考え方のように、最終的な造形にまで視野を広げてデータを扱うことが重要である。

AMFファイルへの埋め込みが可能になったものの、埋め込むためのGcode自体の生成には課題が残る。まず、Gcodeの記述時にノズルとの接触回避や樹脂量の管理などを行う必要がある。こうしたエラーはCADシステムでの設計時に警告したり、スライサ側の処理で修正を行う必要があるだろう。また、埋め込んだ情報に関連するパラメータのやりとりも必要である。XMLタグの属性値として記述しておく方法が考えられるが、どのパラメータを記述しておくべきか、スライサ側で調整をしたい場合はどうするかなどの検討事項がある。

応用的な方法として、実装に使用したGrasshopperとの連携が考えられる。Grasshopperは、さまざまなコンポーネントの組み合わせで、パラメトリックなモデリングを可能とする環境であり、Gcodeに相当するような曲線も作り出すことができる。メッシュの表面から外側に向けて毛を伸ばすように造形したり、適当な位置で樹脂量を制御して模様を描いたりなど、Grasshopperの標準機能を組み合わせることで多様なGcode生成が期待できる。

今後は、ファイルの仕様や生成、スライサでの処理の方法を洗練させるとともに、本手法が有効に用いられる事例を増やし、新たな3Dデータフォーマットとして確立させることを目指す。

7. おわりに

本稿では、工作機械を制御するNCプログラムであるGcodeを3Dモデルデータに埋め込む手法を提案した。熱溶解積層方式3Dプリンタを想定し、AMFファイルのメ

タデータとしてGcodeの情報を埋め込む。この情報をスライサで読み込むことで、最終的なGcodeを生成するために利用できる。具体的な使用例として、Gcodeを直接記述して統合する例とGcode制御を用いたエフェクトのための情報を事前に埋め込んでおく例を示した。AMFファイルを書き出すシステムとスライサソフトウェアを実装することで実現可能性を検証した。

参考文献

- [1] Kanada, Y. Method for Procedural 3D Printing Using a Python Library. *Journal of Information Processing*, Vol. 24, No. 6, pp. 908-916, 2016.
- [2] Laput, G., Chen, X. and Harrison, C. 3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers and Bristles. In *Proc. of UIST '15*, pp. 593-597, 2015.
- [3] LIA. <http://www.liaworks.com/>
- [4] O'Dowd, P., Hoskins, S., Walters, P. and Geisow, A. Modulated extrusion for textured 3d printing. *NIP Digital Fabrication Conference*, No. 1, pp. 173-178, 2015.
- [5] Tanaka, H. and Koda, S. Fabrix - Direct G-Code Editing Tools for Advanced 3D Printing. <https://www.fabrix.design/>
- [6] Hiller, J.D. and Lipson, H. STL 2.0: A proposal for a universal multi-material Additive Manufacturing File format. *20th Annual International Solid Freeform Fabrication Symposium (SFF)*, pp.266-278, 2009.
- [7] ISO / ASTM52915-16, Standard Specification for Additive Manufacturing File Format (AMF) Version 1.2, ASTM International, West Conshohocken, PA, 2016.
- [8] 富士ゼロックス, FAV. <https://www.fujixerox.co.jp/company/technical/communication/3d/fav.html>
- [9] 高橋治輝, 宮下芳明.ブリッジ構造と樹脂の引き伸ばしを用いた高密度な毛構造の造形手法. *日本ソフトウェア科学会 WISS2017*, 2017.
- [10] Takahashi, H. and Miyashita, H. Expressive Fused Deposition Modeling by Controlling Extruder Height and Extrusion Amount. In *Proc. of CHI 2017*, pp.5065-5074, 2017.
- [11] Mueller, S., Im, S., Gurevich, S., Teibrich, A., Pfisterer, L., Guimbretire, F., and Baudisch, P. WirePrint: Fast 3D Printed Previews. In *Proc. of UIST'14*, pp. 273-280, 2014.
- [12] Reiner, T., Carr, N., Mch, R., t'ava, O., Dachsbacher, C. and Miller, G. Dual-color mixing for fused deposition modeling printers. *Computer Graphics Forum*, 33, 2, pp. 479-486. 2014.
- [13] Chen, X., Coros, S., Mankoff, J. and Hudson, S. Encore: 3D Printed Augmentation of Everyday Objects with Printed-Over, Affixed and Interlocked Attachments. In *Proc. of UIST'15*, pp. 73-82, 2015.
- [14] Roquet, C. D., Kim, J. and Yeh, T. 3D Folded Print-Gami: Transforming Passive 3D Printed Objects to Interactive by Inserted Paper Origami Circuits. In *Proc. of DIS'16*, pp. 187-191, 2016.
- [15] Silkworm. <https://projectsilkworm.com/>
- [16] 高橋治輝, 宮下芳明. 熱溶解積層方式3Dプリンタを用いた表現と造形手法のデザインのためのパラメータ探索手法, *インタラクシオン 2018 論文集*, pp. 135-144, 2018.
- [17] Ultimaker Cura (Fuzzy Skin). <https://ultimaker.com/en/products/ultimaker-cura-software>

^{*7} https://en.wikipedia.org/wiki/Model-based_definition