



1. 3Dプリンタで遊ぼう

応
般

水本武志 (京都大学大学院 情報学研究科)
栗野皓光 (京都大学大学院 情報学研究科)
坂東宜昭 (京都大学 工学部)

手軽に使える 3D プリンタ

3次元(3D)プリンタとは、フィラメント状の樹脂を「インク」として、3Dモデルから実際の物体を「印刷」するプリンタです。3Dプリンタを使えば、計算機上で作成した任意の形状の物体をその場で作成できるので、複雑な立体形状のラピッドプロトタイピングに適していると言えるでしょう。

従来の3Dプリンタは非常に高価な設備でした。現在も高精度な3Dプリンタは非常に高価です。このようなプリンタを使うときは、保有する代わりにモデルを送って印刷するサービスを利用するのがよいでしょう¹⁾。一方、近年10万円以下の安価な組み立て式3Dプリンタを販売するスタートアップ、たとえばMakerbot Industries社やPrintrbot社などが現れています²⁾。また4万円程度で完成品を販売する企業も現れました (<http://www.solidoodle.com>)。このようなプリンタとオープンソースの3Dモデリングソフトウェアを組み合わせて、家庭や研究室などで自作モデルの安価な作成が可能となります。たとえば個人的な用途であれば簡単なアクセサリや玩具を、研究用途であれば部品ケースや実験器具、そのマウントなどの試作品を気軽に作れるようになり、実験の幅が広がると期待できます。また、樹脂ではなく食べ物を印刷する3Dフードプリンタなども開発が始まっており³⁾、3Dプリンタはさまざまな場面に広がっています。

本稿では、Makerbot Industries社の3DプリンタThing-O-Maticを題材として、3Dプリンタの組

み立て、モデルの作成、印刷のコツを解説します。特にモデル作成とプリントのコツはほかのプリンタでも活用できるでしょう。なお、Thing-O-Maticは、制御基板にArduino⁴⁾が使われており、フィジカルコンピューティングの一例にもなっています。

プリンタの組み立てとテスト

■ Thing-O-Matic 外観

図-1にThing-O-Maticの外観を示します。225度まで加熱された樹脂射出口に樹脂が送られ、熱で溶けた樹脂が押し出されて印刷台の上に流れ出します。樹脂射出口が下から上に、印刷台が左右と奥行き方向に動くことで、一筆書きの要領で印刷が進みます。

■ 組み立て

まずはMakerbot社からThing-O-Maticの組み立てキットを購入しましょう。マニュアルは付属していないので同社のwikiページを参照して組み立てます (<http://wiki.makerbot.com/thingomatic-doc:thingomatic-assembly-instructions>)。組み立てには多くの工具が必要です。特に組み立ての一部は強い力が必要なので、事前に上記マニュアルを確認してホームセンター等ですべて集めておくのがよいでしょう。

組み立て上の注意点は、次の3つです。

(1) ハンダ付けが必要：ほとんどの配線はコネクタを差すだけですが、一部ハンダ付けが必要です。

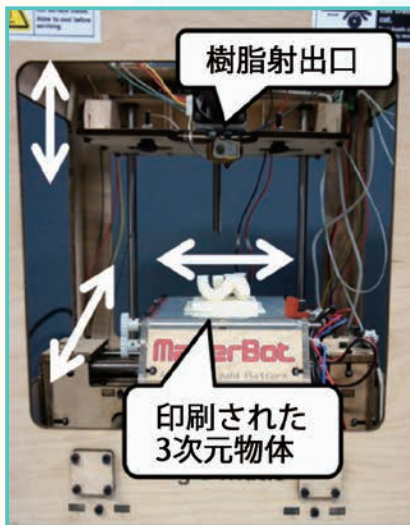


図-1 Thing-O-Maticの外観

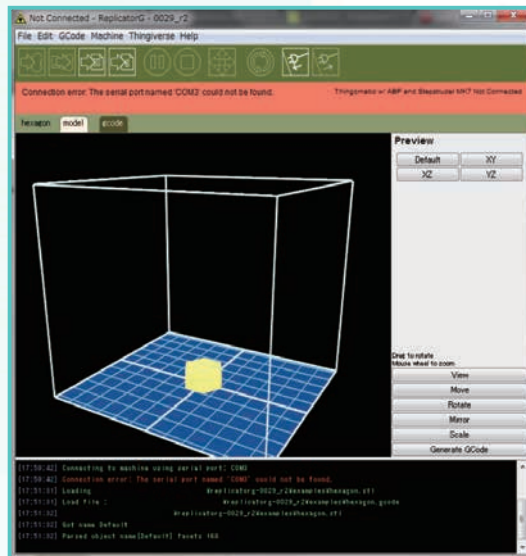


図-2 ReplicatorGの起動画面

- (2) 組み立て時間：作業量が多いので、キットの組み立ては少なくとも丸1日は見積もっておくべきです。
- (3) X StageはAutomated Build Platform：印刷物の土台（印刷台）となるX Stageは、組み立ての難しさと機能とのトレードオフで3種類から選べます。自動温度調整やベルトコンベアによる印刷物の自動取り外しが可能なAutomated Build Platformが印刷のときに便利なので、組み立ては難しいですが、これを作っておけば後でプリントが便利です。

次に、私たちが実際に組み立てる上で気付いたことを箇条書きでまとめていきます。

- (1) フレーム部分の組み立て：プリンタのフレーム部分はベニヤをボルトで固定します。順番に強く締めるとフレームのゆがみが生じてボルトが入らなくなるので、まず全体を緩く固定し、次に順番にボルトを締めましょう。
- (2) ノズル：難所の1つです。ハンダ付けやハサミによる作業が必要なので慎重さが必要です。
- (3) 向きに注意：一見うまく取り付けられたように見えても実は向きが逆であることがあります。マニュアルの写真と十分に見比べて組み立てましょう。

■ ReplicatorGによるプリンタ操作

組み立てが終わったら、次はオープンソースの3Dプリンタ制御プログラムReplicatorGをダウンロードします (<http://replicat.org>, 起動画面は図-2)。この手順も組み立てマニュアルに記載されているので、詳しくはそちらを参照してください。なお、ReplicatorGは、ほかの3Dプリンタにも対応しています。入力ファイル形式は、コンピュータ数値制御（CNC）による機械加工にも使われるGCodeまたは3Dモデルの記述形式の1つであるSTL（Standard Triangulation Language）形式を仮定しています。なお、GCodeとSTLは機械語と高級言語の関係に相当しており、STLファイルを与えるとGCodeへのコンパイルが行われます。

まずソフトウェアの設定を行います。メニューバーのMachine → Machine Typeから最も適切なデバイスを選びます。この設定が間違っても動いてしましますが、成功率と品質が大きく下がるので注意が必要です。続いてコンピュータのUSB端子に3Dプリンタを接続してテスト印刷をします。一般的に3Dプリントは小型かつ安定した形状が簡単なので、ReplicatorGに付属している六角柱のサンプル（hexagon.stl）が適切でしょう。

3Dモデルのコンパイル設定は、Buildボタンをクリックすると表示されるダイアログで選択できま



図-3 3Dプリンタで印刷されたエッフェル塔
(モデルは <http://www.thingiverse.com/thing:15451>)

す。ReplicatorGは、バージョンが上がるにつれて初期値は改善されているので基本的にはデフォルトでよく動きます。ただし以下の4点は注意が必要です。

- (1) Object infill：印刷物の内部充填率を表します。100%なら内部すべてに樹脂を詰めて、0%なら表面だけが印刷されます。100%では印刷時間が長くて樹脂消費量が多くなり、0%では強度が不足しがちなので中間を選択します。
- (2) Feedrate：樹脂の供給速度を表します。モータの速度や印刷時間に影響を与えます。
- (3) Use raft：印刷物の底面に台（ラフト）を追加して印刷するか否かを表します。余分なラフトの印刷によって印刷物を印刷台に強く固定できます。印刷後に印刷台から剥がすのが面倒ですが、ラフトを追加した方が成功率は高まります。
- (4) Use support material：橋などの中空部分を含む物体は、樹脂が下に垂れることがあります。それを防ぐには、この設定を使って下に冗長な部分を追加印刷するのが効果的です。

3Dモデルのダウンロード

ReplicatorGにあるサンプルが印刷できたら、次

はThingiverse (<http://www.thingiverse.com>) から3Dモデルをダウンロードしてより複雑な物体を作ってみましょう。ThingiverseとはMakerbot Industries社が運営するWebサイトで、3Dプリンタのユーザが簡単な置物からスマートフォンのケースまでさまざまな自作モデルをアップロードしています。図-3に、Thingiverseからダウンロードしたエッフェル塔を印刷したものを示します。ライセンスは、Creative CommonsやGPL (GNU General Public License)、BSDライセンス (Berkeley Software Distribution License) など、モデルによって異なるので必要に応じて確認しておきましょう。

ダウンロードしたモデルによっては、位置が中央からずれていたり、大きすぎたりする場合があります。まずは小さなサイズから印刷するのが確実です。というのも、印刷台よりも大きければ当然印刷が不可能である上に、3Dプリンタでは大きな物体の印刷が難しいからです。Rotate、Move、Scaleボタンを用いて3Dモデルの位置やサイズを調整できます。

Thingiverseには多数の便利なモデルがアップロードされています。たとえばcalibrationで検索すれば、大きさや穴の直径が既知のモデルが見つかるので、それを印刷してサイズを測定すれば3Dプリンタのベルトの強さなどの調整ができます。ほかにも、3Dプリンタ名 (CupCake、RepRap、Thingomaticなど) で検索すれば、その製品用のカバーや樹脂のガイドなどの周辺装置も見つかります。このほかにも3DモデルをアップロードするWebサービスはあります。いずれにしてもSTL形式かそれに変換できるファイルを用意すれば印刷ができます。

3Dモデルの設計

次は自分でモデルを作成してみましょう。市販の3Dモデリングソフトウェアを購入し、STL形式で出力させる方法もありますが、本稿ではより手軽なオープンソースソフトウェアによるモデル作成方法を解説します。

OpenSCAD (<http://www.openscad.org>) とは、3次元モデルを作成できるオープンソースソフトウェアです。モデリング専用のスクリプト言語を使い、直方体・球・円柱などの基本的なモデルとそれらの移動・回転、そして和・差・積算等の演算を用いて複雑なモデルが構築できます。このような、基本図形を組み合わせるモデル構成方法は Constructive Solid Geometry (CSG) と呼ばれます。

■ インストール

OpenSCAD の Web サイトから使用するプラットフォームに応じたバイナリやソースコードをダウンロードしてインストールしましょう。たとえば Ubuntu 11.10 の場合、Latest pre compiled Linux x86 (x64) binaries をダウンロードし、展開したディレクトリ中の install.sh を端末で実行します。図-4 に OpenSCAD の起動画面を示します。

■ 基本的な使い方

スクリプトペインに `cube([10, 10, 10]);` と入力し、次にメニューから Design->Compile をクリックし、コンパイルします。するとプレビューペインに1辺が10mmの立方体が表示されます。この立方体をマウスでドラッグするとモデルを回転でき、ホイールを回すと拡大・縮小できます。座標系は View->Show->Axis のチェックボックスを ON にすれば表示されます。

以上のように、OpenSCAD によるモデル設計は、スクリプトを書き、コンパイルし、プレビューペインで確認するという手順の繰り返しとなります。

■ 文法

本稿では基本的な文法のみを解説します。詳しくは WikiBooks の OpenSCAD を参照してください (http://en.wikibooks.org/wiki/OpenSCAD_User_Manual)。文法は比較的簡単なので、手続き型言語の使用経験があれば容易に習得できるでしょう。

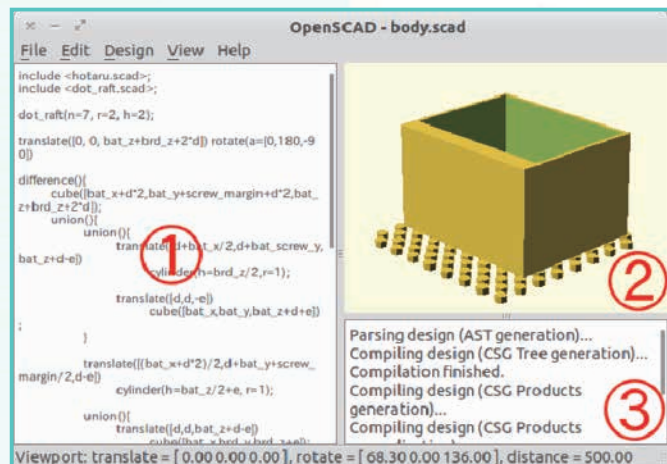


図-4 openSCAD の基本画面：①スクリプトペイン、②プレビューペイン、③ステータスペイン

▶ 基本図形

3Dモデルの構成要素となる基本図形には `cube`、`sphere`、`cylinder` などがあります。ほかにも用意されていますが、これらの組合せだけでも多くのモデルを作成できます。

cube：直方体を表します。寸法は mm 単位で、`x`、`y`、`z` 軸方向の辺の長さを引数にとり、

```
cube([x, y, z]);
```

と書きます。デフォルトではプレビューペインの原点が直方体の頂点に置かれるので、原点をモデルの中央にしたい場合はオプションを追加し

```
cube([x, y, z], center=true);
```

と書きます。

sphere：球体を表します。直径が 10mm の球を作成するには

```
sphere(r=10);
```

と書きます。

cylinder：円柱・円錐を表します。直径 `bar` mm、高さ `foo` mm の円柱を作成するには

```
cylinder(h=foo, r=bar);
```

と書きます。このとき、原点は底面の中心に設定されます。

■ 小特集 夏休み自作自習

引数 r の代わりに $r1$, $r2$ を用いれば、円柱の上底と下底の半径を個別に指定できます。たとえば底面の半径 2mm, 高さ 10mm の円錐は

```
cylinder(h=10, r1=2, r2=0);
```

と書きます。

▶ 移動・回転

基本図形の作成だけでは、すべてが座標系の中心に配置されてしまいます。そこで、基本図形の移動と回転によって配置を操作します。

移動: `translate([x,y,z]) model(...);`

`translate` 文を移動したい基本図形の前に記述して移動量を指定します。引数は x , y , z 軸の各方向への移動量を表します (単位は mm)。`translate` と基本図形の間にセミコロンは不要です。

回転: `rotate([x,y,z]) model(...);`

使い方は `translate` と同じです。引数は、 x , y , z 軸にそれぞれの回転量 (単位は deg) を指定します。

▶ 和・差・積演算

OpenSCAD には 3 種類の演算 `union` (要素の和), `difference` (要素の差), `intersection` (要素の論理積) が提供されており、使用方法は同じです。

```
operator(){ model1(); model2(); ... }
```

これらの組合せだけでも、比較的複雑な 3D モデルが作成できます。たとえば、壁の厚さが 0.1mm で 1 辺の長さが 1mm の箱は次のように書けます。

```
difference(){
  cube([1,1,1], center=true);
  translate([0,0,0.1])
  cube([0.8,0.8,1], center=true);
}
```

▶ 変数定義・モジュール化

OpenSCAD は多くのプログラミング言語と同様、変数定義、モジュール化、制御構文をサポートします。

変数: OpenSCAD では、宣言なしに変数を定義できます。それらの加減乗除や論理演算など基本的な演算もできます。たとえば、壁の厚みを変数 d で

表すために $d=1$; を先頭で定義しておけば、前記の立方体の箱は次のようにも書けます。

```
d=1;
difference(){
  cube([1,1,1], center=true);
  translate([0,0,d])
  cube([1-d*2,1-d*2,1], center=true);
}
```

3D プリントでは壁の厚みの調整などの試行錯誤が頻繁に生じます。したがってパラメータの調整を簡単にしておくことは重要です。

モジュール化: 一般的な関数定義と同様に、次のように複数の命令をモジュール化できます。

```
module mod_1(arg1, arg2, ...){
  model();
}
```

モジュール化すれば基本図形と同様に呼び出せるので、複雑な形状を再利用するときに便利です。

ほかにも、`for` 文や算術演算を組み合わせることで複雑な曲線を持つモデルなども作成が可能です。

3D プリントのコツ

3D プリント失敗の最も多い原因は、目詰まりと `Curling` です。目詰まりとは、プリント中に樹脂射出口が詰まって樹脂が射出されなくなることであり、一方 `Curling` とは、3D プリンタの印刷台と印刷物の接触する部分が少しずつ剥がれていく現象のことです。図-5 に `Curling` が起こった印刷物の写真を示します。

目詰まりの原因の多くは、射出口についた樹脂のカスの取り除き忘れなどの整備不良を原因とする射出用モータへの過負荷です。一方で、`Curling` による印刷物の浮き上がりによる射出口の圧迫が原因の目詰まりも多く起こります。したがって、まず `Curling` の防止が成功率向上の近道でしょう。

`Curling` を防ぐには次の 5 点に注意しましょう。

(1) z 軸初期値: 3D モデルの最底面を印刷するときの射出口の高さです。これを下げれば印刷台に樹脂が密着するので `Curling` が生じにくくなり

- ます。ただし、近づけすぎると射出口が圧迫されて目詰まりの原因になるので調整が必要です。
- (2) 移動速度：射出機構と印刷台のモータ速度です。移動速度を下げることで位置精度が上がり、プリントの精度が上がります。プリントの精度が上がれば印刷物体の歪みが軽減され、Curlingを防ぐ効果も期待できます。
 - (3) ラフトの形状：印刷物体の印刷台への密着度を高めるための台です。ReplicatorGでUse raftを選択すると長方形のラフトを自動的に生成しますが、円形のラフトなら力が分散するので最もCurlingを防ぐ効果が高くなります。したがって、3Dモデル自体に円形ラフトを追加し、Use raftを選択せずに印刷するとよいでしょう。
 - (4) 射出口の温度：デフォルトの温度よりも少し高く設定すると粘度が下がるので密着度が上がり、目詰まりの改善にもつながります。上げすぎると樹脂が焦げますが、私たちの経験では235度くらいまでは上げて印刷できます。
 - (5) 印刷台の温度：上げてはいけません。私たちの経験では、印刷台のベルトコンベアを構成する樹脂テープが溶けて破損してしまいました。

3Dプリンタで遊ぼう

本稿では、安価な3Dプリンタを用いた自作3Dモデルの印刷方法を解説しました。現状では印刷精度は高価な製品には劣る上に、美しい印刷には試行錯誤が必要なので、誰でもすぐに印刷できるという段階ではありません。しかし、任意の形状の物体が

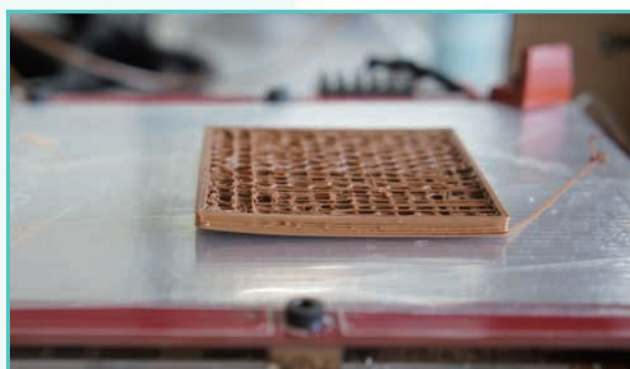


図-5 左手前部分が反ったCurlingの例

その場で手に入るので、使いこなせば汎用的に使える非常に便利なツールとなる可能性を持っています。この夏に遊んでみてはいかがでしょうか。

参考文献

- 1) 日曜に考える, 日本経済新聞 (29 Jan. 2012).
- 2) Wallich, P.: Taking Share, IEEE Spectrum, pp.57-58 (Jan. 2012).
- 3) Periard, D., et al.: Printing Food, Proc of the 18th Solid Freeform Fabrication Symposium, Austin TX, pp.564-574 (Aug. 2007).
- 4) 菅野 創: Arduino 入門, 情報処理, Vol.52, No.8, pp.922-925 (Aug. 2011).

(2012年4月16日受付)

水本 武志 (学生会員) | mizumoto@kuis.kyoto-u.ac.jp

2010年京都大学大学院情報学研究所情報学専攻修士課程修了。同年同専攻博士課程進学。学振特別研究員DC 2。主に人とロボットの合奏と野外生物の合唱の研究に従事。

粟野 皓光 (学生会員) | awano@easter.kuee.kyoto-u.ac.jp

2010年京都大学工学部卒業。同年同大学院情報学研究所情報学専攻進学。2011年通信情報システム専攻へ転専攻。現在在学中。半導体プロセス微細化に伴う特性ばらつきに関して、チップの試作、解析技術、モデル化の観点から研究している。

坂東 宜昭 | yoshiaki@zeus.kuis.kyoto-u.ac.jp

2010年京都大学工学部に入學。2011年より京都大学大学院情報学研究所奥乃研究室でAndroidアプリの開発などに従事。

