

## 03

3D プリンタもプログラミングで  
— draw3dp

金田 泰 (Dasyn.com)

## 概要

現在主流になっている3D設計や3D印刷の方法は汎用性があるが万能ではないから、ほかの方法が必要なこともある。3Dプリンタでオブジェクトを造形するとき、通常はそのオブジェクトの3DモデルをCADソフトウェアとマウスなどを使用して設計する。また、設計したモデルをスライサというソフトウェアをつかって3Dプリンタが入力できる形、つまりGコードに変換する。モデルにあたえられる情報は基本的にその表面形状だけであり、それだけでは十分でないことがある。また、この方法ではうまく印刷できない形状もある。そのため、従来の3Dモデルの拡張やほかの方法による設計・印刷がもとめられる場合がある。

この記事ではこのような際に有効な3D設計・印刷法の拡張や新たな方法を紹介する。つまり、モデル上の各点で方向(印刷方向)を指定できる場指向オブジェクト・モデルや、手続き的なプログラムを使用した設計法、水平方向に限定されない印刷法などを紹介する。これらの方法は主流の方法が持つ汎用性はないが、それが適する目的たとえば中空立体の造形においてはそのちからを発揮する。この3Dモデルおよび設計・印刷法の概要については論文<sup>1)</sup>で説明し、設計・印刷に使用するライブラリ draw3dp についても論文<sup>2)</sup>に記述したが、この記事ではこの方法の背景、関連動向、応用などを紹介する。

## 従来の3D印刷とGコード

3D印刷の普及にともなって従来の3D設計の方法も広く知られるようになってきたが、設計されたオブジェクトがどのようにして3D印刷されるかはま

だ十分に知られているとはいえないだろう。そこで、まず簡単にそれを説明する。

3Dプリンタを使用して3次元のオブジェクトを造形するとき、通常は次のような手順で設計・印刷する。まず3DCADを使用して宣言的なモデルを設計し、そのモデルをスライサと呼ばれるソフトウェアで水平にスライスして、多数の薄い層を生成する。3Dプリンタにはさまざまな種類があるが、安価なタイプはFDM(Fused Deposition Modeling, 熱溶解積層)型と呼ばれ、とかしたフィラメント(プラスチック)をノズルの先端から射出してかためる(図-1)。そのため、スライスして得られた層をさらに細いフィラメントに分解し、その印刷データをコマンド列(手続き)のかたちでプリンタにおくって印刷する。

CADが出力するファイル形式はさまざまだが、スライサにおくるにはSTL(Standard Triangulation Language または Stereo-Lithography)という宣言的な形式のファイルが使用される。STLはモデルの表面形状を3角形の集合によって近似する。

3D印刷するときには、加工の手続きをGコードという言葉によって記述する。3D印刷は付加加工(additive manufacturing, AM)という機械加工の一種とされていて、機械加工を制御するためのプログラムが必要になる。この目的のために使用されるのがGコードというアセンブラ風の言語である。Gコー

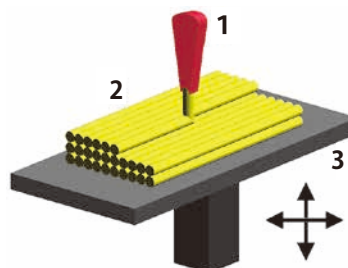


図-1 FDM型3Dプリンタの原理  
("FDM by Zureks" by Zureks - Wikimedia Commons)

ドは元々工作機械の刃の動作をあらわすので手続きの(動的)である。

3Dプリンタは通常は層ごとに印刷するが、Gコードは層という概念に制約されてはいないから、3Dプリンタのヘッドは実はもっと自由に動かせる。スライスした結果は通常Gコードによって表現され、それによって印刷ヘッドの動作やプラスチックの射出速度などが指定される。3Dプリンタでは通常、水平にスライスされた層ごとに印刷するので、層間の移動のとき以外はヘッドが垂直方向に動くことはない。しかし、Gコードは層にしばられていないので、ヘッドは常に上下にも動かせる。

## 新しいデザイン法とモデル

今後3D設計や3D印刷をさらに発展させるであろう新しいデザイン法と新しいモデルを紹介する。

### ジェネラティブ・デザイン

現在では、コンピュータをデザイン(設計)の中核で使用する新しいデザイン法としてジェネラティブ・デザイン、あるいはアルゴリズムック・デザインという手法が浸透してきている。3D印刷においては、オブジェクトを造形するとき通常はその表面の形をCADソフトウェアとマウスなどを使用して設計する。この方法ではその形を設計者自身が決める。これに対して、造形するオブジェクトの形をプログラムや数式によって決めるやり方がジェネラティブ・デザインである。このデザイン法を採用する目的はさまざまだが、機械設計においては最適化(たとえばトポロジー最適化)が目的であることが多い。

「ジェネラティブ」ということばはデザインよりもグラフィクスや音楽などのアートとむすびつけられている。ジェネラティブ・デザイン<sup>3)</sup>もジェネラティブ・アート<sup>4)</sup>もプログラムや数式を使用して生成されるが、特にグラフィクスにおいてはProcessingというプログラミング言語と強くむすびついている。ジェネラティブ・アートとは、コンピュータ・プログラムや数式のような「遺伝子」から生成されるアートのことである。ジェネラティブ・

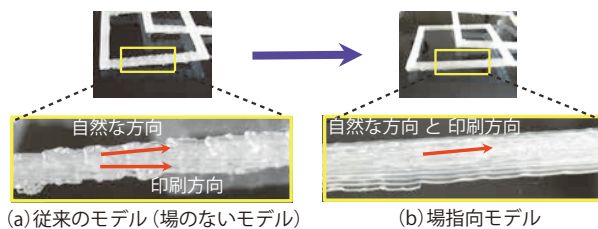


図-2 2つのモデルにおける自然な方向と印刷方向との関係

デザインもそういう「遺伝子」によるデザイン(設計)のことを意味する。

フラクタルやカオスがそれを生み出す数式そのものからは想像できない形を生成するように、プログラムが生成するデザインはそれを実行してみて(「遺伝子」が発現して)はじめてその姿をあらわす。形状がプログラムによって自由に決められるときはもちろんだが、建築のようにおよその形が事前に決められているときでも、その細部はプログラムを実行することによってはじめて決められる。

### 場指向オブジェクト・モデル

この節ではモデルの拡張の例として場指向オブジェクト・モデル<sup>1)</sup>を紹介する。このモデルは方向づけられていない従来の3Dモデルの問題点を解決するために筆者が提案した。

従来の3D印刷法においては印刷の手順は完全に自動的に決められる。そのため、印刷方向はモデル化するオブジェクトが持つべき「自然な方向」と矛盾することがある。特に通常は印刷の方向が常に水平なので、自然な方向が水平でないときには矛盾が起こる。この矛盾はいわゆる「階段効果」つまりオブジェクト表面を階段状にする原因になり、造形されたオブジェクトの形状も強度も劣化する(図-2)。

これに対してdraw3dpが立脚しているモデルにおいては、オブジェクトは磁場のような「場」の中にあって、印刷の向きはオブジェクト上の各点でのこの場の向きから決まる(図-3)。このモデルを場指向オブジェクト・モデルと呼ぶ。印刷の手順(手続き)はオブジェクトの形を決めるプログラム中で決められる。これによって自然な方向と印刷方向とを一致させ、階段効果を防止する。

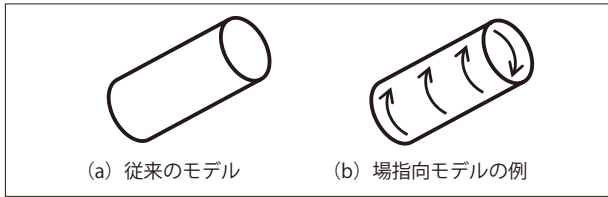


図-3 従来のモデルと場指向モデルによる円柱の例

## 手続き的な 3D 設計と 3D 印刷

新しい設計法・印刷法の 1 つである手続き的な 3D 設計と 3D 印刷について述べる。新しい印刷法と書いたが、実はコンピュータによる機械加工の歴史は手続き的な方法からはじまっているので、まずその歴史から説明する。

### 手続き的な切削加工の歴史

コンピュータ数値制御 (CNC) による切削加工の技術とそのための言語 APT<sup>5)</sup> はすでに 1940 ~ 1950 年代に開発されている。数値制御 (NC) の技術は 1942 年に John T. Parsons によって発明されたが、CNC の技術はそれにもとづいて 1950 年代にマサチューセッツ工科大学 (MIT) において開発された。この制御のため、MIT においては APT が開発され、切削加工のプログラムを記述するのに使用された。APT は基本的にはアセンブラ風の言語であり手続き的だった。

しかし、その後、CAD 技術が開発され、APT のような手続き的な方法によって設計者が指示することはほとんどなくなった。設計者は CAD によって静的つまり宣言的なモデルを記述し、それをコンピュータが手続き的なプログラムに変換するようになった。そうになったのは、切削加工には手続き的記述が適さない複雑さがある上、G コードや APT は低水準で抽象化機能が弱いため設計者がプログラムするのは困難だからだろう。

### Python ベースの 3D 設計・印刷用言語

場指向オブジェクト・モデルをジェネラティブに設計・印刷するため、Python のライブラリ draw3dp.py を筆者が開発した<sup>2)</sup>。その開発の第 1 の目的は従来の CAD などのソフトウェアではあつかえない場指向オブ

ジェクト・モデルをあつかえるようにすることである。従来のソフトウェアの単純な拡張ではそれを実現できない。

第 2 の目的は、このモデルにもとづくジェネラティブ・デザインのためには Processing のようなプログラミング言語を使用するのが適切であるから、それをこのライブラリによって実現することである。Processing ではなく Python を使用するのには、Python の方がプログラミング言語としての機能がより整っているからである。

第 3 の目的は、前記のように切削加工においては成功しなかった手続き的な方法を付加加工 (3D 印刷) において成功させることである。切削加工と付加加工の性質に違いがあるために、後者においては成功する可能性があると考えられる。つまり、付加加工は切削加工より直観的なので設計者がそれほど困難なく手続き的に記述することができ、それによって宣言的な記述からはうまく印刷できないモデルでもうまく印刷できる場合があるだろうということである。

開発されたライブラリの概要を説明する。draw3dp.py は 3D 印刷を手続き的に記述するための Python のライブラリであり、Web 上で公開されている。プログラミング言語としては当然の機能だが、Python が持つ手続き抽象化機能 (つまり関数) を使用すれば G コードではできないモジュラーな 3D 印刷が実現される。このライブラリを使用すると部品の組立てと変形を行ってオブジェクト・モデルを生成し、それを G コードに変換できる。Python ベースの言語といっても、Python に 3D 印刷用のライブラリ (API) を追加したものである。

### 「糸」にもとづく部品の表現

場指向オブジェクト・モデルにおいてはオブジェクト上の各点での場の方向が決められているだけだが、draw3dp.py においては印刷コマンドへの変換を容易にするため、オブジェクトの各部分の印刷の順序が決められている。

FDM 型などの 3D 印刷においては糸状の材料 (FDM 型においてはフィラメントという) をかさね

て造形する。そこで、draw3dp.pyにおいては部品を太さのある直線状の糸 (string)  $S_i$  のならび ( $S_1, S_2, \dots, S_n$ ) として表現する:

$$S_i = (P_{i\text{start}}, P_{i\text{end}}, c_i, v_i)$$

ここで  $P_{i\text{start}}$  は糸の始点であり  $P_{i\text{end}}$  はその終点である。  $c_i$  は糸の断面積,  $v_i$  は印刷速度 (秒速) つまりヘッドの移動のはやさである。  $v_i$  は概念的には不要だが、実装 (糸の印刷) には便利なパラメタである。  $v_i$  を適切に指定することによって、従来の宣言的な方法では印刷できなかった形状がこの手続き的な方法では印刷可能になることがある。たとえば、後述する中空の球は特に極付近での印刷速度を適切に決めることによって、うまく印刷できるようになる。糸のならびはオブジェクトの表現だが、上記の表現はオブジェクトが手続き的に生成されることを反映している。

この表現によって部品の各点におけるフィラメントの方向が規定され、厚みのある部品は内部のフィラメントの構造や密度も記述される。これらはいずれも従来のCADやSTLでは記述できない。このような表現を選択した本来の意図は、前記のように3D印刷におけるフィラメントの方向を印刷されるオブジェクトの表現 (美観など) に活かすことだった。

## 部品の組立て

部品は単純な部品を組み合わせる (追加していく) ことによって構成する (図-4)。つまり、まず空の部品を作る。現在ライブラリで用意している単純な部品としては円、平面螺旋、ヘリックス (立体螺旋) などがある。たとえば helix というメソッドは1重のヘリックスつまり底面がない円筒をえがく。層なしにかつ層のつぎめもなしに、任意の高さのうすい円筒をえがくことができる。これらの単純な部品を上記の空の部品に追加していくことによって、オブジェクトの形を作る。

また、これらの単純な部品組合せでは作れない形状もさらに単純な部品をあらゆる低水準の関数の組合せで記述できる。つまり、直線をひいたり (GコードのG1コマンドに相当) 指定された点までフィラメントを射出せずに移動したり (G0コマンドに相当) し

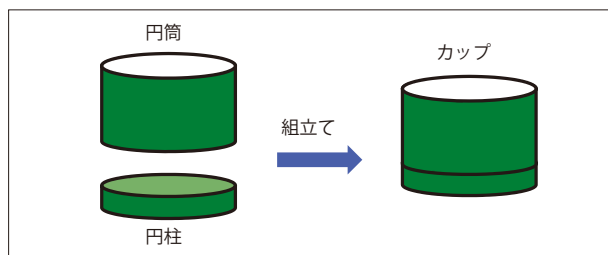


図-4 部品の組立ての例

て形を作る。糸の断面積や印刷速度も低水準の関数を使用して制御する。

部品を組み合わせる際には、組み合わせた部品が印刷可能になるための条件をみとす必要がある。つまり、上記のライブラリを使用するとき、すでに印刷したフィラメントによって部品の印刷が妨害されず、そのフィラメントや印刷台 (プリントベッド) によってあとから印刷するフィラメントが支持されるという条件をみとす必要がある。この条件がみとされれば、部品を逐次に印刷することで組み立てられる。部品の印刷順序をどのように決めてもこの条件がみとせなければ部品を分割し印刷順序を変更してみとすようにする<sup>1)</sup>が、これは常に可能なわけではない。

部品の組立て手続きをPythonの関数として表現すれば複合部品を生成する関数となり、モジュラーな構造 (プログラムの構造および印刷された部品の構造) が実現される。

## オブジェクトの変形

部品やその組合せを自由に変形 (deform) させられれば、特に非線形変換を使用すれば比較的容易に多様な形状が作れる。したがって、draw3dp.pyでは変形のための関数が用意されている。従来のCADでも並進、回転、拡大、縮小などの線形変換が用意されているが、コンピュータ・グラフィクスにおいてはそれだけでなく変形が重視されている。draw3dp.pyにおいても変形によって複雑な形状が作れるようになっている。

変形という操作の利点は多様な形状が生成できることだけではなく、3D印刷可能な部品を用意しておけば変形後も印刷可能性を維持しやすいという利点もある。すなわち、ジェネラティブ・デザインに

よる 3D 印刷が成功するためにはそのデザインが多数の制約をみたく必要がある。しかし、部品の組立てだけでそれらを見たくのは容易でない。そのため、変形を併用して、より容易にそれらを見たくせるようになっている。

変形のための関数として直交座標によるものと円筒座標によるもののが用意されている。これらの 2 つの関数の機能は基本的に等しいが、2 個あるのは直交座標の方が容易な場合と円柱座標のほうが容易な場合とがあるからである。これらの変形関数は糸の始点と終点の座標を変換する。

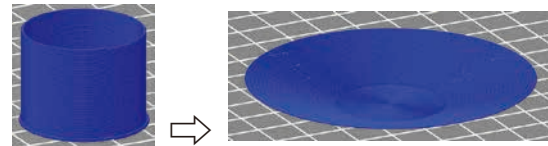
ヘリックスをはじめ、変形前の部品をそのまま印刷すると印刷ヘッドは近似的には水平に移動する。しかし変形によってヘッドを上下に波打たせることもできる。その効果についてはこの記事の末尾で言及する。

変形の例を図-5 に示す。Repetier Host という 3D 印刷ツールを視覚化のために使用している。図-5(a) はヘリックスとすいすい円柱(底)とで構成されるカップとそれを変形して得られた図形を示す。カップのヘリックスの部分を変形して皿の形状にし、底のサイズはヘリックスを変形して得られた図形にあわせる。図-5(b) はヘリックスとそれを変形して得られる図形の例として球を図示している。ここでは変形時にフィラメントのピッチが保存される(つまりヘリックスを縦方向には伸縮せずに球にまきつける変形をする)。

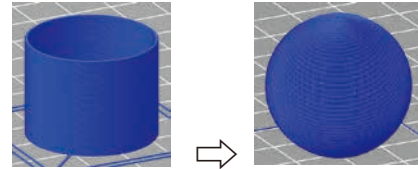
### 印刷の変調によるテクスチャマップ

この方法ではオブジェクトの表面をノズルで走査するようにして造形していくが、この走査線によってテレビの画像をえがくようにオブジェクトの表面に文字、画像、テクスチャなどをえがく(テクスチャマップ)ことが比較的容易にできる。つまり、フィラメントの断面積を変化させて凹凸を作る。この方法を適用することを印刷を変調する(modulate)という。印刷中にフィラメントの断面積を変化させれば、あまり大きな凹凸は作れないが、比較的こまかい凹凸を作ることができる。

フィラメントの断面積を変化させるにはフィラメントの射出速度を変化させるか印刷ヘッドの移動速度を



(a) カップ形状から皿形状への変形



(b) ヘリックスから球への変形

図-5 変形の例

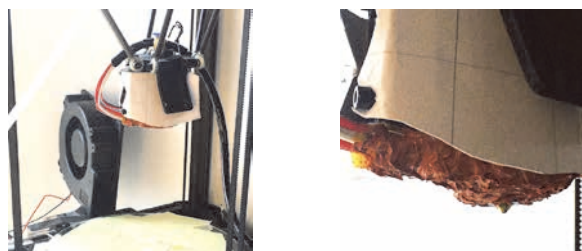
変化させればよいが、より応答性が良い後者の方法によって変調している。3D プリントにおいてはフィラメントを押し出す押し出し機(extruder)の動作の変化がヘッド先端のフィラメントの動作に反映されるまでの遅延が大きく、場合によっては数秒かかるため前者の方法は応答性が悪い。そのため、応答性が良い印刷ヘッド移動速度を変化させる方法を使用している。

## 手続き的な 3D 印刷の実践

従来 3D 印刷は主にプロトタイプ開発に使用されてきたが、最近は航空機部品をはじめ 3D 印刷された部品を製品に使用する機会が増えている。「螺旋 3D 印刷」と呼ぶ手続き的な 3D 印刷の実践においても、低コストで印刷できる利点を活かして製品に適用することを筆者はめざしている。

### 印刷環境

螺旋 3D 印刷のために執筆者が実現した環境について、まず簡単に説明する。G コードを使用することから分かるように特殊な環境は必要ないが、第 1 にできるだけヘッドを上下に大きく移動させながら印刷できる環境の実現をめざしている。現在は 2 万円程度の安価なキットから製作したデルタ型のプリンタを使用しているが、G コードによって制御できればほかのタイプのものでもよい。デルタ型を選択したのは、モデルの変形によってヘッドが大きく上下するようになって対応できると考えたからである。だが、ヘッドを回転させられない通常の 3 軸の 3D プリンタは実際にはあまり大きく上下させられない。それ



(a) 冷却用ラジアルファンと印刷ヘッド (b) 冷却防止対策を施したヘッド先端

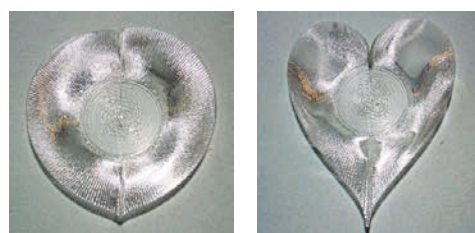
図-6 デルタ型3Dプリンタを使用した印刷環境

は、すでに印刷した部分との衝突を避ける必要があるからである。しかし、現在では多数5～6軸のプリンタが試作され製品化されたものもある。それらが使用できれば、ヘッドが大きく上下するようになってうまく印刷できるだろう。

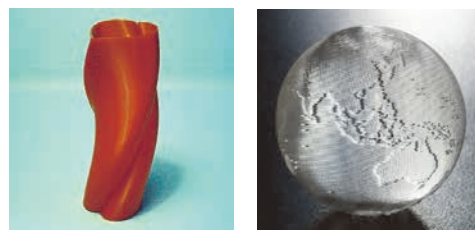
第2に、螺旋3D印刷においては高速に、10～30分程度で1個のオブジェクトが低コストで印刷できるようにするため、印刷後のフィラメントを強力に冷却するとともに、ヘッドの冷却を避けるようにしている(図-6)。フィラメントの冷却のためには図-6(a)のように直径12cmの強力なラジアルファンを使用している。このファンは印刷ベッド付近に固定しているため、高さのあるオブジェクトを印刷する際には工夫が必要になる。また、ヘッドの冷却をさけるためにヘッドの周囲を300℃の高温にも耐えるシリコンゴム(RTV)によっておおっている(図-6(b))が、それでもノズル先端は冷却されるため、PLAでも通常より高温の250℃程度を指定して印刷している。Webで検索すると冷却のためにクロスフロー・ファンを使用した例や固定したファンからダクトでノズル付近まで風を導く例があるが、そこまで高速性を追求した試行は限定されている。

### さまざまな形状

螺旋3D印刷によって印刷した容器などの例を図-7(a),(b)に示す。図-7(a)はヘリックスを図-5(a)のようにひらたく変形して作った皿だが、ヘリックスにハート型曲線の方程式<sup>☆1</sup>にもとづく変形を適用している。このようにフィラメントを水平に近い角度でまきつけるときもサポートなしで



(a) ハート型の皿



(b) ねじったハート形の花瓶 (c) 変調した球としての地球儀  
図-7 さまざまな形のオブジェクト

きる。場所によって皿の角度やフィラメントの密度がことなるために光の反射に変化が生じるが、透明度が高いPLA(ポリ乳酸)を使用すると写真のような輝きが得られる。図-7(b)は同じハート形を使用した花瓶である。

すでに説明したように、ヘリックスを変形して球を作ることができる。印刷時に球を1点だけでは支持できないため支持に少し工夫が必要だが、通常の意味のサポートを使用する必要はない。この球にビットマップによる変調の技法を使用して作ったオブジェクトを示す。図-7(c)が示しているのは透明な球を地図によって変調して作った地球儀である。地球儀などの印刷の様子はYouTube<sup>☆2</sup>で観察できる。

### LEDとの組合せ—ランプシェード

螺旋3D印刷はうすくて透明かつ光を拡散するオブジェクトを作るのに適しているため、そのオブジェクトをLEDと組み合わせて照明器具にする(つまりランプシェードを螺旋3D印刷によって作る)のが有力な応用だと考えられる。透明な地球儀もランプシェードとして使用できる。従来の3D印刷にもとづくランプシェードのジェネラティブ・デザインも多数試行されている(たとえばJ. Kytönenら<sup>3)</sup>)。これらはみな不透明なシェードであるのに対して、螺旋3D印刷では透明度が高いシェードが作れる。

☆1 山本信雄: [http://www.geocities.jp/nyjp07/heart/index\\_heart.html](http://www.geocities.jp/nyjp07/heart/index_heart.html)

☆2 <https://www.youtube.com/user/ykanada>

透明なシェードを通して目に入る光も美的だが、シェードを通過する光が作り出す図-8のような陰も美しい。このような光も陰も FDM 方式による螺旋 3D 印刷だからこそ実現されるものであり、射出成形で同じ形を作っても実現できない。この陰の形はシェードの形やテクスチャだけでなく、フィラメントを重ねる際にどれだけ上下にくねらせるかによっても変化する。場指向モデルが活かせる応用だといえる。

MIT の J. Klein らはガラスの 3D 印刷を実現し、その透明さを活かすために光源を置き、美しい陰を作っている<sup>☆3</sup>。材質や陰の形は違うが、透明な 3D 印刷品から魅力をひきだすために共通の方法を使っているといえる。

実用的な照明器具を作るには技術者・研究者だけでなくデザイナーとコラボレーションするのがよいと考えられる。筆者は螺旋 3D 印刷のこのようなプロジェクトを実施している。以下コラボレーションをめざす理由を述べる。ジェネラティブ・デザインによるランプシェードはプログラムによって生成されるが、プログラムを書いたりその入力を決めたりするのは人間である。また、そのランプシェードを持つ照明器具をデザインするのも人間である。このようなデザインには知識やスキルが必要であり、それを持つプロフェッショナルはデザイナーであるから、プログラムを記述する技術者・研究者とデザイナーとがコラボレーションするのがよい。

また、これとは逆の方向としてアマチュアによる照明器具デザイン・製作も試行している。すなわち、3D 印刷の良いところはある程度の手助けがあれば消費者が自分でオブジェクトや器具を製作できる（つまり Maker になれる）ことであるから、このような方向のために照明器具製作のワークショップも行っている。

## 展望

3D 印刷の応用分野は広がり、設計手法としてもジェネラティブ・デザインが使われるようになってきているが、3D モデリングの方法はまだ限られている。汎用的なモデリング法があらゆるタイプの造形



図-8 3D 印刷シェードが作る陰

に使用できるならそれでよい。しかし、現在普及している 3D 設計・印刷の技術はより高い強度を得るための機械設計のために発展してきたものであり、強度より美観を重視する用途など、ほかの用途には必ずしも適していないと考えられる。3D 印刷においてもっとさまざまな方法が試されてよいはずである。筆者が試行している手続き的なプログラミングにもとづく方法もその 1 つだが、今後もっと多様な方法が試され、そこから、これまでにない造形物が登場することを期待したい。

### 参考文献

- 1) Kanada, Y.: 3DPrinting of Generative Art Using the Assembly and Deformation of Direction-specified Parts, Rapid Prototyping Journal, Vol.22, No.4 (2016).
- 2) Kanada, Y.: Method for Procedural 3DPrinting Using a Python Library, Journal of Information Processing, Vol.24, No.6, pp.908-916 (2016).
- 3) Bohnacker, H., Gross, B., Laub, J., and Lazzaroni, C.: Generative Design, Princeton Architectural Press (2012), 日本語版, ビー・エヌ・エヌ新社 (2016).
- 4) Pearson, M.: Generative Art: A Practical Guide Using Processing, Manning Publishing Co. (2011).
- 5) Ross, D. T.: Origins of the APT Language for Automatically Programmed Tools, ACM SIGPLAN Notices, Vol.13, No.8, pp.61-99 (1978).

(2017年2月26日受付)

金田 泰 (正会員) ■ yasusi@kanadas.com

1981年日立製作所入社後、中央研究所、システム開発研究所、カーネギーメロン大学、RWCP つくば研究所等において Fortran コンパイラ、ベクトル記号処理と論理型言語処理、創発的計算のモデル、情報抽出/検索/組織化、仮想環境型コミュニケーション、ネットワークとポリシー、ネットワーク仮想化、深層学習によるコンピュータビジョン等の研究開発に従事し、2016年退職。博士(工学)。工学院大学情報学部非常勤講師。

☆3 <http://bit.ly/1KwmgFv>, <http://bit.ly/2kBMtn3>