

3D-printing of Generative Art by using Combination and Deformation of Direction-specified 3D Parts

Yasusi Kanada
Dasyn.com, Japan
yasusi@kanadas.com

Direction-specified 3D modeling and FDM-based printing methods enable expression of natural directions, such as hairs, fabric, or other directed textures, in modeled objects. This paper describes a method for creating various shapes of generative artistic objects with several specialized attributes by applying three new techniques to the direction-specified methods for better artistic expressions. The most important technique is “deformation”, which enables deforming simple 3D models to create varieties of shapes much more easily in generative design processes. The second technique is called the spiral/helical printing method, which enables consistent print-direction vector field, i.e., filament directions, of the surface consistent with those of the interior portion and enables seamless or less-seam printing results. The third technique controls light reflection while printing by using the spiral/helical printing method with transparent PLA. It enables the printed objects reflect light brilliantly. The proposed method with these three techniques was implemented as a Python library and evaluated by printing various shapes, and it is confirmed that this method works well and objects with attractive attributes, such as the brilliance, can be created.

1. INTRODUCTION

Conventional 3D design and printing methods cannot express and embody models with internal attributes and some surface attributes. An important attribute is direction [Kan 14] [Son 13]. 3D Printing methods such as fused deposition modeling (FDM) can express a direction at each point in the model, including each internal point, by the printing direction. Print heads of 3D printers usually move in horizontal directions, but they can express non-horizontal directions by moving and stacking strings (i.e., filament) toward non-horizontal directions. Several advantages are given by non-horizontal printing. First, so-called staircase effect [Son 13] can be avoided by non-horizontal head motion. This effect weakens the mechanical strength of the object and it also makes the object ugly. Second, expressing direction is important especially when the purpose is production printing rather than prototyping because objects with natural or artificial directions, such as the direction of hair or grass, can be expressed in a better way [Kan 14].

A methodology for printing 3D objects with specified printing-direction by using the FDM method was originally proposed by a poster [Kan 13]. This methodology, which is called the direction-specified 3D design and printing methodology, was described in detail and a method based

on this methodology was implemented in the previous paper [Kan 14]. Users can select predefined directed parts, give parameters, assemble them, and print the assembled model by using this method.

The major purpose of this method is to design and to print 3D artistic models rather than industrial models generatively [Kan 14] or algorithmically. This method is also intended to use for producing final products but not for producing prototypes.

This paper proposes a method for generatively designing and printing direction-specified 3D models that is based on the above methodology with the following three new techniques. First, “deformations” of models are proposed; that is, this paper extends the direction-specified methodology by adding another step called deformation to enable generating varieties of shapes much more easily by a generative design process. Unlike computer graphics, preservation of 3D printability is important for deformation in direction-specified 3D printing. Second, the spiral/helical printing method is proposed as a method for direction-specified 3D printing methodology. This method enables consistent print-direction vector field, i.e., filament directions, of the surface consistent with those of the interior portion, enables seamless or less-seam printing results, and enables low-angle overhang. Third, a method for controlling light reflection while using the spiral/helical printing method is introduced. By using the proposed method, the printed objects can reflect light brilliantly.

The rest of this paper is organized as follows. Section 2 outlines the proposed designing and printing method. Section 3 describes the method of deformation. This section includes the history of deformation and a description of the 3D printability concept and a method for preserving it in the deformation process. Section 4 describes the method for spiral/helical printing. Section 5 explains the method for light-reflection control. Section 6 explains an implementation and evaluations using simple combinations and deformations of parts such as a helix (an empty cylinder) and a (filled) cylinder, which generate shapes of dishes, vases, wine grasses, and so on. Section 7 concludes this paper.

2. OUTLINE

This section outlines the proposed methodology for printing a 3D object with specified printing direction.

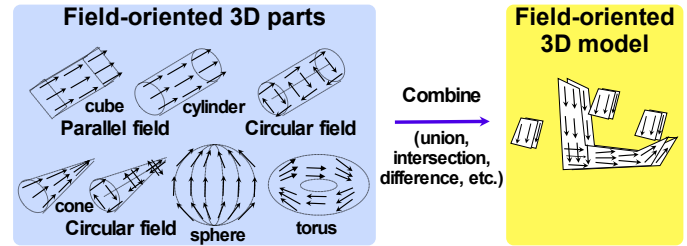
The purpose of this methodology is to design and to print generative art objects, so generative design is first

explained. There are two types of 3D CAD tools, i.e., manual-design tools and generative (or algorithmic) design tools, and both types of tools can be used for both free-form design and design by parts assembly. Manual-design tools are used for manually specifying parts and models using pointing devices. Many commercial and free manual-design tools, such as AutoCAD [Aut 14], SolidWorks [Sol], or Blender [Ble] are available. Models can be created as free-form models, i.e., freely designed by drawing lines or other shapes by using manual-design tools by pointing devices, or created by assembling predefined parts and specifying parameters manually. In contrast, generative-design tools generate models or specify parts algorithmically by using a declarative or procedural language, such as Processing [Pea 11]. Therefore, the user of the tools, i.e., the programmer, writes a program to generate a model. Several generative-design tools, such as OpenSCAD [Wik], are available.

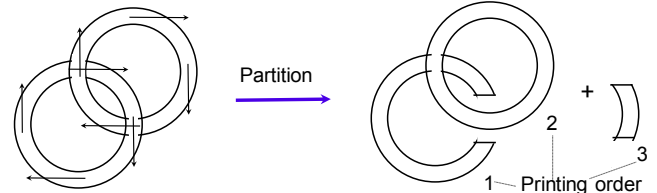
The design and printing process of the proposed method consists of five steps: field-oriented modeling, partitioning, deformation, field-based tool-path generation, and non-horizontal 3D printing (Fig. 1) [Kan 14]. The first three steps are the design steps. These steps are briefly reviewed below using Fig. 2.

In the field-oriented modeling step, an object model with vector field is created. This model is represented by a modeling language, which can express a vector field. It is an extension of conventional solid model. A direction is defined at each (3D) point in the object as a vector. A field-oriented model can be created by assembling parts in a palette (Fig. 2(a)) [Kan 14]. Although the palette in this figure contains various parts, some of them are composite parts and still abstract (i.e., the tool-path is not yet specified).

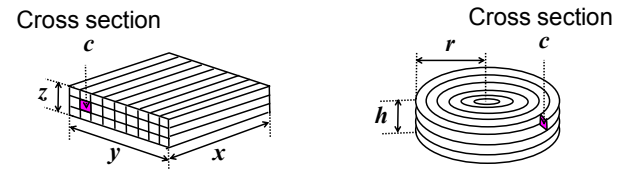
In the partitioning step, the designed model is partitioned to a list of strings (fragments) that can be printed. This step consists of two sub-steps, *printability enhancement* and *peeling* (or *hashing*). Printability enhancement, which is



(a) Field-oriented modeling by parts combination



(b) Printability enhancement sub-step of partitioning step



(c) Peeling (hashing) sub-step of partitioning step

Fig. 2 Partitioning a model for printability enhancement

explained in the previous paper [Kan 14] so it is not explained here, is applied only when the model becomes printable by dividing to two or more sub-models (see Fig. 2(b) for example). In the peeling sub-step, the model is “peeled” (“hashed”) along the field vectors (Fig. 2(c)) and a set of *strings* are generated as the result. These strings can be regarded as filaments but they may be still abstract; that is, concrete tool-path and printing attributes might not yet been determined. Each string S_i can be represented by

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

where $P_{i_{\text{start}}}$ means the start point of the string, $P_{i_{\text{end}}}$ means the end point of the string, c_i means the cross section of the string, and v_i means the printing speed. v_i is conceptually unnecessary but it is useful in implementing (printing) the string.

The “peeled”-model generation process is completely different from processes used in conventional 3D printing, which is based on the Standard Triangulated Language (STL) and slicing. If the field vectors are not in parallel, peeling follows them; that is, the directions of strings are not necessarily horizontal. Advanced methods for hashing are described in the previous paper [Kan 14].

The execution order of the design steps, i.e., the first three steps, is conceptually shown in Fig. 1; however, the order is not strict. One possible method is to assemble non-partitioned parts into a field-oriented solid model first, to partition it then, and to deform it. Another possible method is to select and to deform pre-partitioned parts and to assemble them into a field-oriented solid model. In this

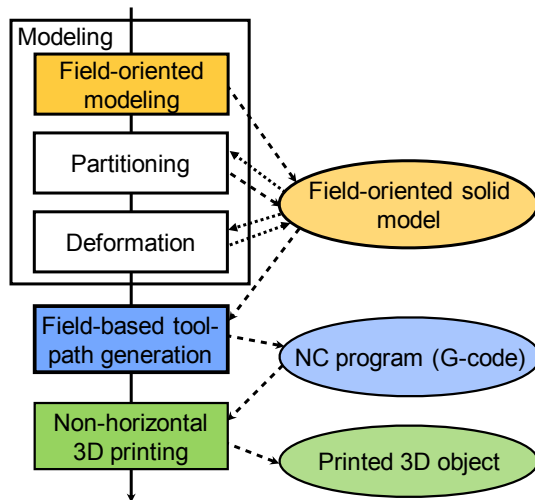


Fig. 1 Design and printing process for direction-specified 3D printing

case, the first step is partitioning, which is included in part design process, and the second step is deformation.

In the deformation step, the designed model can be deformed to various shapes. The detail is described in the next section.

In the field-based tool-path generation step, a tool-path is generated as an NC program such as G-Code. Not only the orbit of the print head, but also the amount of extrusion, is calculated for each string. A tool-path is generated by ordering strings. The detailed method is described in the previous paper [Kan 14].

Finally, in the printing step, the numerical-control (NC) program is executed by a 3D printer and a printed 3D object is outputted.

3. DEFORMATION

In this section, first, a problem concerning deformation in 3D printing, which does not exist in graphics, and a solution of the problem are described. Second, the printability preservation concept is described.

3.1 Deformation method for 3D printing

This paper extends the direction-specified methodology by adding another step called “deformation” to enable generating varieties of shapes much more easily in generative design processes. Although deformation has not been discussed in 3D-printing context, this paper focuses on generative 3D design and printing, which is based on a deformation-based method.

To preserve printability of the model, deformation for 3D printing requires controlling two attributes of strings that consist the peeled model. They are cross section and printing velocity. This requirement is 3D-printing specific. The 3D-printability concept is explained and the preservation issues are explained in the next subsection.

Two types of deformations are defined. One is Descartes-coordinate based deformation and the other is cylinder-coordinate based deformation. Both translate coordinates, cross section, and printing speed.

Describing a deformation using Descartes coordinates is sometimes useful, so the following function is defined in the library.

`deform_xyz(fd(x, y, z), fc(c, x, y, z), fv(v, x, y, z))`

In this specification, function *fd*(*x*, *y*, *z*) (i.e., the first argument) maps a location (*x*, *y*, *z*) to a new location (*x*₁, *y*₁, *z*₁), so it returns three values. Function *fc*(*c*, *x*, *y*, *z*) (i.e., the second argument) maps a cross section at location (*x*, *y*, *z*) to a new cross section at (*x*₁, *y*₁, *z*₁). Function *fv*(*v*, *x*, *y*, *z*) (i.e., the third argument) maps a printing speed at location (*x*, *y*, *z*) to a new speed at (*x*₁, *y*₁, *z*₁). Function *fc* must be monotonically increasing about *c* and function *fv* must be monotonically increasing about *v*. Theoretically, any mapping can be specified for *fd*, *fv*, and *k*. However, at least they must be continuous and smooth to get a collect printing result.

It would be better if the cross section could automatically be optimized; however, it is currently difficult. So the cross section must be manually specified in the method proposed in this paper.

Cylinder coordinate is more useful for describing a deformation, especially when axisymmetric models are deformed, so the following function is defined in the library.

`deform_cylinder(fd(r, θ, z), fc(c, r, θ, z), fv(v, r, θ, z))`

In this specification, function *fd*(*r*, *θ*, *z*) (i.e., the first argument) maps a location (*r*, *θ*, *z*), which is expressed in cylinder coordinates, to a new location (*r*₁, *θ*₁, *z*₁). Function *fc*(*c*, *r*, *θ*, *z*) (i.e., the second argument) maps a cross section at location (*r*, *θ*, *z*). Function *fv*(*v*, *r*, *θ*, *z*) (i.e., the third argument) maps a head speed at location (*r*, *θ*, *z*) to a new speed. The same monotonicity conditions as *deform_xyz* exist for *fv* and *fc* for *deform_cylinder*. Functions *fd*, *fv*, and *fc* must be continuous and smooth.

Examples of deformation are visualized in **Fig. 3**. (A 3D-printing tool called Repetier Host was used for visualization.) Figure 3(a) shows a cup, which consists of an empty cylinder and a thin cylinder (i.e., the bottom). This cup can be transformed to a dish shown in Fig. 3(b) by applying the following deformation to the empty cylinder part.

`deform_cylinder(fdd(r, θ, z), fcd(c, r, θ, z), fvd(v, r, θ, z))`
 where *fdd*(*r*, *θ*, *z*) = (*r* + 1.05 *z*, *θ*, 0.3 *z*). (1)

The forms of functions *fvd* and *fcd* will be described in the implementation section because they depend on the techniques described in Sections 3.3 and 4. The thin cylinder, i.e., the bottom, must be resized to fit to the deformed empty cylinder.

The cup can also be transformed to a vase shown in Fig. 3(c) by applying the following deformation.

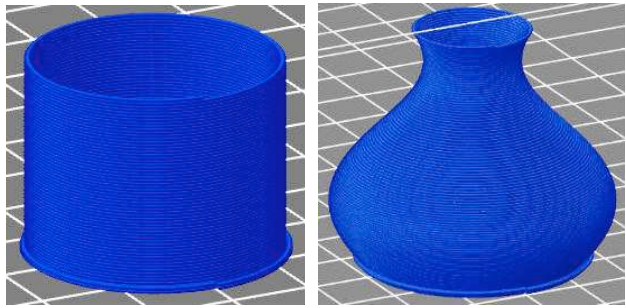
`deform_cylinder(fdp(r, θ, z), fcp(c, r, θ, z), fvp(v, r, θ, z))`
 where *fdp*(*r*, *θ*, *z*) = (*r* (0.8 + 0.4 sin(*z* / 8 + 6.5)), *θ*, *z*). (2)

Fig. 3(d) shows an empty cylinder (without a bottom). This cylinder can be transformed to a sphere shown in Fig. 3(e), which preserves the filament pitch, by applying the following deformation.

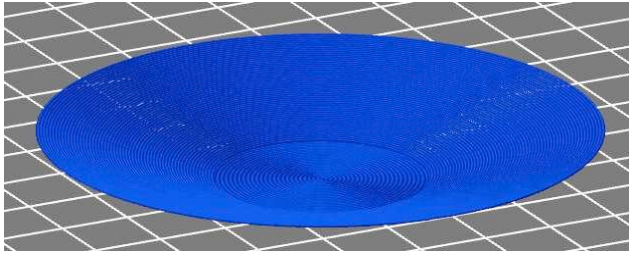
`deform_cylinder(fds(r, θ, z), fcs(c, r, θ, z), fvs(v, r, θ, z))`
 where *fds*(*r*, *θ*, *z*) = (*Radius* * sin(*π z* / *cylinderHeight*), *θ*, *r* - *Radius* * cos(*π z* / *cylinderHeight*)) (3)

Parameter *cylinderHeight* is the height of the empty cylinder before the deformation and it is half of the equator length of the sphere generated by the deformation.

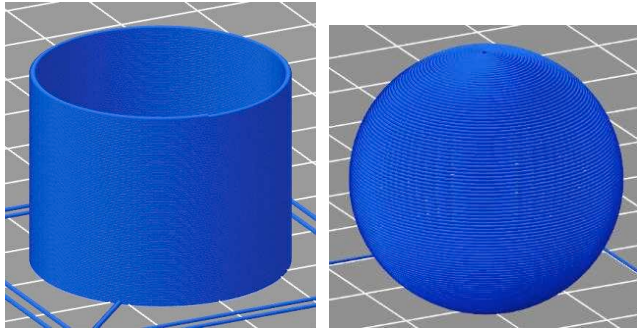
Moreover, non axisymmetric shapes such as shown in Fig. 3(f), which can be generated by deforming axisymmetric shapes shown in Fig. 3(a) and (d), can be easily generated by using cylinder-coordinate based deformations. Although various shapes exist in Fig. 3, all these shapes are generated only using these trigonometric functions.



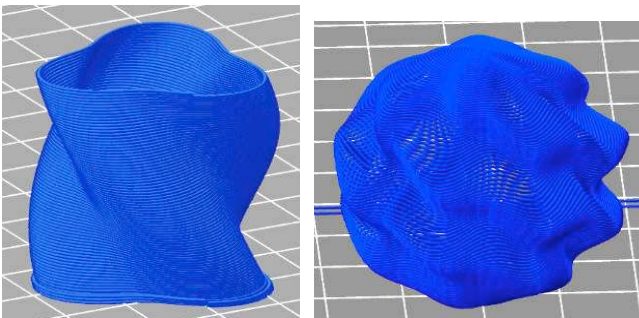
(a) Cup – original shape (c) Vase – deformation result



(b) Dish – deformation result



(d) Empty cylinder – original shape (e) Sphere – deformation result



(f) Non-axisymmetric shapes – deformation results

Fig. 3 Examples of deformation

However, other types of functions can of course be used.

3.2 Printability preservation

The concept of 3D printability plays an important role to avoid printing failures in the proposed method. By giving the 3D printability measure to each string of a peeled model and by managing the measure while deforming the model, the printability of the model can be preserved. That means, if the deformation makes the string density sparser, the cross section must be increased to preserve the printability. If the amount of extrusion is constant, new string may fail

to stack and drop (Fig. 4(a)). On the contrary, if the deformation makes the string density denser, the cross section must be decreased. Otherwise, excess filament may cause bumps because of the elasticity of printed filament (Fig. 4(b)). A more detailed explanation of printability preservation will be described in a future paper.

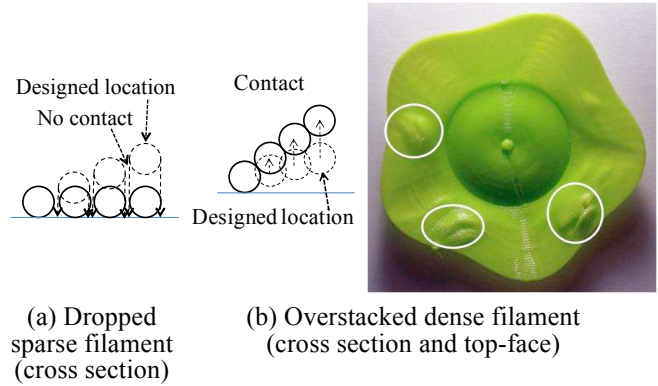


Fig. 4 Printing failures

4. SPIRAL/HELICAL PRINTING METHOD

The *spiral/helical printing method* enables printing axisymmetric direction-specified 3D models such as cylinders or spheres with consistent print-direction vector field, i.e., filament directions, and without seams or with less seams. The spiral/helical printing method means a method for printing an object spirally or helically with adjusting cross section and printing speed to preserve the 3D printability (Fig. 5). If a model to be printed (or a part in assembled parts) is axisymmetric, it is natural to give a circumferential-direction vector in each location in the model. Such a model is peeled to be a collection of circumferential strings, and a natural method for printing such a model is to print it spirally or helically. This method enables reducing non-printing head motions; that is, enables more seamless printing. Seamlessness matters when printing final products or artistic objects. To print a widening or narrowing helix (i.e., a helix with increasing or decreasing radius) by this method, cross section and printing speed, must be properly controlled as described in the previous section. In addition, this method can also generate thin and strong structure.

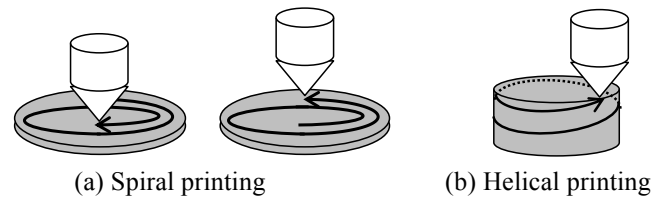


Fig. 5 Spiral/helical printing method

By using the spiral/helical printing method, a quickly widening helix such as a dish-like shape or a narrowing helix such as an umbrella-like or dome-like shape can be printed without support material. That means, this print method allows high-angle overhangs. Conventional 3D-printing methods do not allow high-angle overhang, e.g., more than 75 degrees, without support material. Because

support material causes several problems including extra material consumption and difficulty in removing it, it is beneficial if no support is required. Although the range of shapes that can be printed by this method are restricted, it is still beneficial.

However, when printing a widening or narrowing helix at a higher speed, the cross section should be decreased or increased to stabilize printing (to preserve printability). Otherwise, printing failures such as shown in Fig. 4 may occur.

5. LIGHT-REFLECTION CONTROL

Some material used for FDM printers reflects light on the surface and brilliantly shines, and the amount and the direction of reflection can be controlled by certain methods. This is an attractive attribute for artistic or visual-design purposes. For example, transparent polylactic acid (PLA), especially pure PLA, is quite attractive in reflection (see Fig. 8); that is, reflection becomes strong in 3D-printed object made of PLA because strings increase the surface area that reflects light. This attribute is caused not only by the surface but it is affected by the internal structure of the object. If the filament is not transparent, the brilliance disappears. The reflection can be controlled by overhang angle, filament density, or some other attribute of the spiral/helical printing method (**Fig. 6**). Fig. 6(a) shows reflection controlled by the overhang angle. If the angle of the light is changed, the different portions of the object more strongly reflect the light. Fig. 6(b) shows reflection controlled by filament density. Even if the angle of the light is changed, the dark part never reflects light brilliantly.

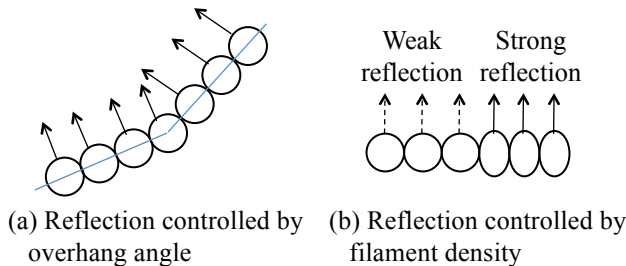


Fig. 6 Reflection control

6. IMPLEMENTATION AND EXAMPLES

6.1 Python library

A preliminary version of a modeler program library, **draw.py**, which is publicly available at http://www.kanadas.com/program-e/2014/10/-3d_printing_library_for_parts.html, for generating G-Code for 3D printers was developed and used for evaluation. By using this library, the user can write a program to select parts, such as a line, a helix, a cylinder, to specify parameters of the parts, and to combine them. In the current version, deformations can only be applied to the combined model, but it will be applied to parts or partially combined model too.

Model data representations in a program can be described as follows. A peeled model is represented by a list of strings in this library program as described in Section 2. A

string is a named tuple (or structure) that contains coordinates x , y , and z , a cross section, and a printing speed.

A model part can be generated by a function defined in the library. For example, the following function generates a string list that represents a cylinder.

```
draw.cylinder(radius, height, vpitch, hpitch, x0, y0, z0)
```

Parameters “radius” and “height” determines the shape of the cylinder, parameters $vpitch$ (i.e., vertical string-pitch) and $hpitch$ (i.e., horizontal string-pitch) determines the cross section, and $x0$, $y0$, and $z0$ determine the bottom center coordinates of the cylinder. Parts can be concatenated (i.e., assembled) to form an object.

Operations defined in the library can be described as follows. A Descartes-coordinate based deformation functions directly on these values. A cylinder-coordinate based deformation translates the Descartes coordinates, i.e., x , y , and z , to a cylinder coordinates, and functions on the translated values, and translates the results back to a Descartes coordinates in the current version of the library. This method may be optimized in a future version.

6.2 Creation of dishes and vases

Various dishes and vases can be created by deforming a cup shown in Fig. 3(a), i.e., a combination of an empty cylinder and a thin cylinder. The results of printing a basic form of dish shown in Fig. 3(b) and several variations by Rostock Max 3D-printer are shown in **Fig. 7(a)** to (c). The results of printing a basic form of vase shown in Fig. 3(c) and several variations of deformed cups are shown in Fig. 7(d) to (f). The dish and vase were created by deformations using horizontal and/or vertical sine waves. Deformation techniques described in Section 3 were applied to both.

A dish shown in Fig. 7(a) was printed by deformation (1) in Section 3.2. The translation functions for printing speed and cross section used for deformation described in Section 3.2 are as follow.

$$fcd(c, r, \theta, z) = 0.96 c$$

$$fvd(v, r, \theta, z) = v$$

These functions qualitatively follow the techniques described in Sections 3.3 and 4; however, the functional forms and constant values were defined experimentally.

Figure 7(b) shows a variation of dish (with 6-cycle modulation) and Fig. 7(c) shows a dish. The deformation function contains $\cos(4\theta)$, which generates 4-cycle patterns. The deformation function uses the light-reflection control technique. Moreover, the photos in this figure show the dish with two light-source directions. The brighter areas move while changing the light-source direction.

A plain vase shown in Fig. 7(d) was printed by deformation (2). The translation functions are as follows.

$$fcp(c, r, \theta, z) = 1.9 c$$

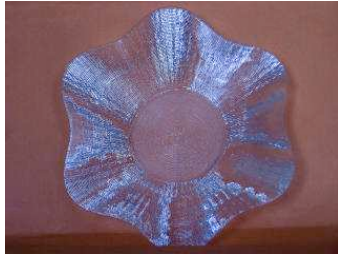
$$fvp(v, r, \theta, z) = (1 + 0.4 \sin(z/4) \cos(3\theta)) v$$

Figure 7(e) shows a skewed cup and Fig. 7(f) shows a

“wine glass”. They are generated from a cup too. The shapes of all the dishes and pods are approximated by 72 linear lines per round trip of the head. The turning points of the head and filament can be observed in these photos, especially in Fig. 7(a) to (c). The dishes will look better if they consist of finer lines, but it will take more time to print them.



(a) Plain dish



(b) 6-cycle dish



(c) Reflection-controlled dish with two light directions



(d) Plain vase



(e) Skewed cup



(f) Wine glass



(g) Sphere

Fig. 7 Dishes and pods generated by various deformations

6.3 Creation of spheres

A sphere can be created by deforming a thin cylinder shown in Fig. 3(d). An example of printed sphere is shown in Fig. 7(g). The printing speed must be reduced and the cross sections of strings must be decreased in the lower and upper parts of the sphere. The amount of increase and decrease was still determined experimentally. A set of translation functions are as follows.

$$fcs(c, r, \theta, z) = 1.2c$$

$$fvs(v, r, \theta, z) = 1.2((fr(r, \theta, z) / \text{Radius})^2 + 0.1)v$$

where fr is the first component of fds (i.e., the radius). Printability preservation techniques for spiral/helical printing method are not applied to this set of functions. However, if the value of fvs is increased, that is, the printing speed is optimized, some techniques including the stabilization of printing are required.

7. CONCLUSION

This paper describes a method for creating various shapes of generative artistic objects with several specialized attributes by applying three new techniques to the direction-specified methods for better artistic expressions. The first technique, i.e., “deformation”, enable deforming simple 3D models to create varieties of shapes much more easily in generative design processes. The second technique, i.e., the spiral/helical printing method, enables generating thin and strong structure and low-angle overhang. The third technique controls light reflection and enables the printed objects reflect light brilliantly. The proposed method with the first two techniques was implemented as a Python library and evaluated by printing various shapes.

By this implementation and evaluation, two conclusions are confirmed. The first conclusion is that the proposed method works well for thin axisymmetric shapes. The second conclusion is that object with variety of shapes and attractive attributes such as brilliant reflection can be obtained by deforming thin axisymmetric shapes.

REFERENCES

- [Aut 14] Autodesk, <http://www.autodesk.co.jp/products/all-autocad>
- [Bar 84] Barr, A. H., “Global and local deformations of solid primitives”, *ACM SIGGRAPH Computer Graphics*, Vol. 18, No. 3, pp. 21–30, July 1984.
- [Ble] <http://www.blender.org/>
- [Coq 90] Coquillart, S., “Extended free-form deformation: a sculpturing tool for 3D geometric modeling”, *ACM SIGGRAPH Computer Graphics*, Vol. 24, No. 4, pp. 187–196, September 1990.
- [Kan 13] Kanada, Y., “A Method of 3D Printing which is Consistent with Natural Direction in Shape”, Poster, *International SFF Symposium 2013*, August 2013.
- [Kan 14] Kanada, Y., “Method of Designing, Partitioning, and Printing 3D Objects with Specified Printing Direction”, *2014 International Symposium on Flexible Automation (ISFA 2014)*, July 2014.
- [Pea 11] Pearson, M., “Generative Art: A Practical Guide Using Processing”, Manning Publishing Co., 2011.
- [Sed 86] Sederberg, T.W. and Parry, S. R., “Free-form deformation of solid geometric models”, *ACM SIGGRAPH Computer Graphics*, Vol. 20, No. 4, pp. 151–160, 1986.
- [Sol] <http://www.solidworks.com/>
- [Son 13] Xuan Song, Yayue Pan, and Yong Chen, “Development of a Low-Cost Parallel Kinematic Machine for Multi-Direction Additive Manufacturing”, *SFF Symposium 2013*, August 2013.
- [Une 08] Unemi, T., Matsui, Y., and Bisig, D., “Identity SA 1.6: An Artistic Software that Produces a Deformed Audiovisual Reflection Based on a Visually Interactive Swarm”, *2008 International Conference on Advances in Computer Entertainment Technology (ACE '08)*, pp. 297–300, 2008.
- [Wik] “OpenSCAD User Manual”, WikiBooks, http://en.wikibooks.org/wiki/OpenSCAD_User_Manual