Support-less Horizontal Filament-stacking by Layer-less FDM

Yasusi Kanada

Dasyn.com, Tokyo 164-0013, Japan

Abstract

When using conventional additive manufacturing (AM) methods, material is stacked vertically and layer-by-layer. These methods cause two problems; that is, an object with overhang or skewed stacking structure is difficult to be created by these methods without support material and "seams" are easily generated when transiting between layers. This paper proposes a layer-less fused-deposition-modeling (FDM) method, which enables mostly horizontal stacking of filament without support material and which can avoid seams easily. Such filament-stacking is enabled by increasing the height of the print head gradually, i.e., without layer transitions that make horizontal stacking difficult. The proposed method also supports techniques for controlling printing directions and various printing-direction-dependent expressions, such as fiber-like textures or brilliance, and "deformation" and "modulation" techniques, which enables generation of various shapes and textures. These techniques make AM products attractive as arts or as final products for consumers. Objects to be printed by the proposed method can be modeled as directed solid models designed by a component-based method (i.e., a new CAD based method) or a generative method, which are completely different from conventional CAD based methods.

1. Introduction

When using conventional AM methods, material is stacked vertically and layer-by-layer; however, these methods cause two problems. One problem is that an object with overhang or skewed stacking structure, such as a shallow plate (dish) or an empty sphere, is difficult to be created by these methods without support material. Conventional 3D printing methods do not allow low-angle overhang, e.g., more than 75 degrees. The other problem is that "seams" between layers, such as shown in **Figure 1**, are easily generated. By using a conventional 3D printing method, A model to be printed is first sliced into thin layers and they are printed one by one. A seam is generated when transiting from one layer to the next layer. Such seams are highly visible when printing a thin object such as a plate. Although a shallow plate can be mostly correctly printed [Kan 15d] as summarized in Section 2 by using special techniques with a conventional design and printing method, seams caused by layer-transitions are difficult to be avoided by this method. A better method should be developed.

This paper proposes a layer-less fused-deposition-modeling (FDM) method, which is called the helical/spiral printing method, enables mostly horizontal stacking of filament without support material and also enables printing without seams. Such filament-stacking is enabled by increasing the height of the print head gradually, i.e., without layer transitions that make horizontal stacking difficult. This method was originally introduced and explained in a top-down manner by another presentation [Kan 14b]; however, this paper describes the design and manufacturing process in a bottom-up manner, focusing on the above two issues.

The proposed method allows controlling printing directions and various printing-direction-dependent expressions, such as fiber-like textures or brilliance. The direction-specified printing

1

methodology was originally proposed by the author [Kan 13][Kan 15b]. There are several advantages to direction-specified printing. First, the so-called staircase effect [Son 13] can be avoided by non-horizontal head motion. This effect weakens the mechanical strength of the object and also makes the object look ugly. Second, direction expression make AM products attractive as final products for consumers or as some kinds of industrial products; 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, and the details of objects can be expressed in a better way [Kan 14a]. In addition, a technique for controlling light reflection while using the helical/spiral printing method can be introduced. By using this technique, the printed objects can reflect light brilliantly.
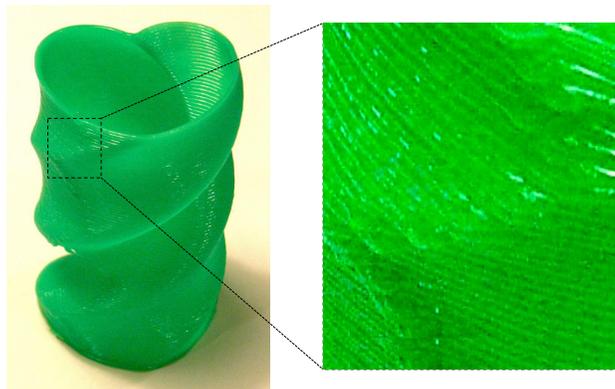

Figure 1: Seams between layers

The proposed method can also include two techniques called *deformation* [Kan 15a] and *modulation* [Kan 15c] to enhance the shapes and textures of printed objects. An object to be printed by this method is represented (modeled) by a sequence of "strings". The first process, "deformation", can be applied to this representation to generate various shapes. The second process, extrusion "modulation" or texture mapping, can also be applied to objects of this representation to generate textures, or characters or pictures, on the surfaces of objects.

The proposed method is suited for bottom-up (generative or emergent) design and artistic applications rather than top-down (intentional and artificial) design and industrial applications. That is, this method can be used for a completely different design method, i.e., generative design method. Generative-design tools generate models or specify parts algorithmically using a language such as Processing [Pea 11]. Generative design is more suited for artistic objects.

Although conventional modeling tools cannot be used for the direction-specified method, a new component-based modeling method can be applied to it. That means, objects to be printed can be modeled as directed solid models designed by a component-based method (i.e., a new CAD based method) or a generative method.

The rest of this paper is organized as follows. Section 2 describes a trial of support-less printing of shallow and thin plates by using a conventional method, which cannot avoid the problem of seams. Section 3 describes the method for helical/spiral printing that solves both problems. This section also includes descriptions of modulation and light-reflection control techniques. Section 4 describes the method of deformation and shows print results of deformed and modulated objects. Section 5 proposes a CAD-based method for direction-specified modeling and manufacturing, which include the helical/spiral printing process. Section 6 concludes this paper.

2. Support-less Printing of Shallow Plates by a Conventional Method

A technique for printing objects with overhang without support by using a conventional 3D-printing method is described in this section. Shallow thin plate is chosen for an example. It was not possible to eliminate seams caused by layer transitions while using the conventional method. A more detailed description is available in another paper [Kan 15d].

A thin plate can be designed and visualized by conventional 3D modeling and printing tools. **Figure 2**(a) visualizes G-code that represents a plate only and Figure 2(b) visualizes G-code that represents a plate and support. The support in Figure 2(b) can be smaller, but it may still sometimes use more material than plate itself. (To visualize G-code, a 3D-printing tool called "Repetier Host" (Macintosh version) was used for these figures.)

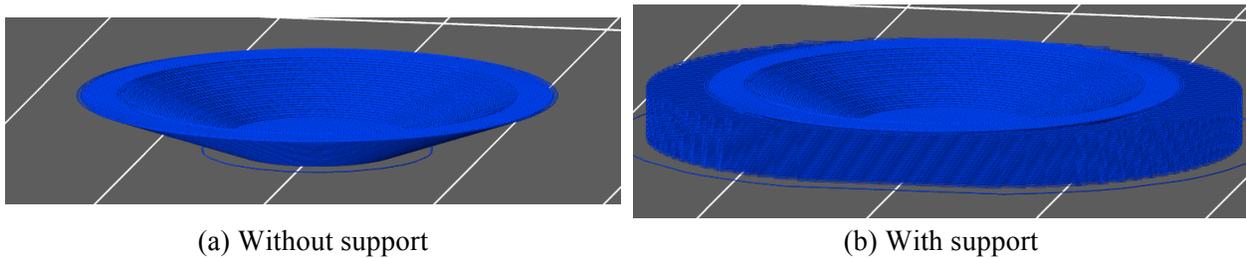| (a) Without support | (b) With support |

Figure 2: Visualized G-code of plate with overhang

The above plate was designed by OpenSCAD [Wik 15]. A plate was represented by a tapered "cylinder" with different sizes of top and bottom surfaces. The design (program) is omitted here, but mostly the same program is publicly available.[1] By executing this program, a plate with less than 0.2 mm thickness is generated. It is so thin because each layer consists of only a single wind of filament. However, the thickness of printed plate will be more than 0.2 mm because it is equal to the filament diameter, which is expected to be 0.4 mm as described later. OpenSCAD can generate an STL (Stereolithography or Standard Triangulated Language) file.

To enable printing a shallow plate without support, two configurations of the slicer must be modified. First, the amount of filament should be increased. Four to six times larger amount of filament should have been specified for obtaining successful results. Second, by changing the configuration, the layer height should be adjusted. It is usually smaller than the diameter of the nozzle, which extrudes melted plastic. If the diameter is 0.5 mm, the normal layer height is, for example, 0.4 mm. However, the layer height is set to smaller value here. The layer height can be 0.1 mm.

The reason why a layer height should be smaller is explained as follows. **Figure 3** shows part of a cross section of a plate. This figure assumes that the filament is wound with the angle of 15° from the horizontal surface. If the diameter of the filament is 0.4 mm (or slightly larger), the difference of the heights of the neighboring filaments is 0.4 sin 15° (= 0.1) mm. The layer height should thus be 0.1 mm. However, because the horizontal filament location that the slicer chooses may be different from this figure, the layer height may have to be adjusted.

Figure 4 shows part of the printed plates with 15° gradient round by using slicers called Slic3r (a) and Skeinforge (b). The layer height was set to 0.15 mm, which was optimized by cut-and-try, is much larger than the designed value, 0.1 mm, which is the expected value for 0.4-mm-diameter filament. The amount of filament is four times as much as the original value. This

---

[1] http://www.kanadas.com/weblog/2015/02/cad.html or http://bit.ly/173UG6b

causes the diameter much larger (the measured diameter was 0.7 mm). Both slicers mostly worked well, but actually the filaments draw more exact concentric circles in the case of Slic3r.

There are several problems in these results. The most serious problem in these printed plates is the layer seam. Especially in Figure 4(b), several gaps that spoil the plate can be observed around the seams, but they can also be observed in Figure 4(a).
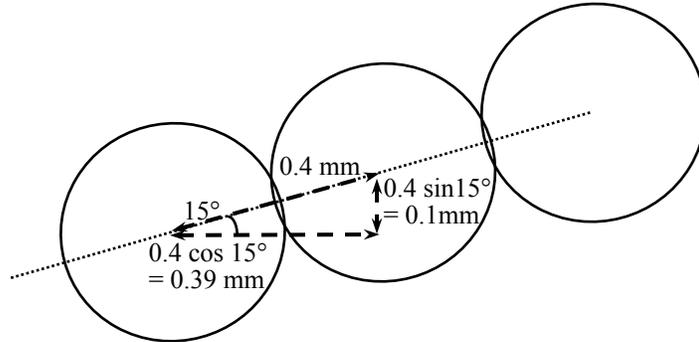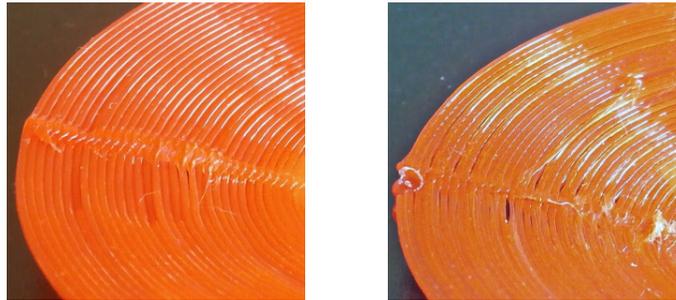

Figue 3: Decision of layer height

(a) When using Slic3r   (b) When using Skeinforge

Figure 4: Layer seams of printed plate (layer height: 0.15 mm)

3. Helical/spiral Printing Method

The helical/spiral printing method, the representation of object models for this method, and the texture mapping and the light reflection techniques used with this method are explained in this section. This method solves the problem of seams described in the previous sections.

3.1 Basic helical/spiral printing method

The proposed 3D-printing method called "helical/spiral printing method", which enables seam-less printing of thin objects is described in this subsection. By using transparent material such as pure polylactic acid (PLA), light, reasonably strong objects, which reflects light brilliantly, can be printed by this method.

The helical/spiral printing method enables printing axisymmetric direction-specified 3D objects such as thin cylinders or spheres, and enables printing without seams or with less seams. The helical/spiral printing method means a method for printing an object spirally or helically (**Figure 5**). This method enables non-printing head motions to be reduced; that is, it enables printing with less seams. Seamlessness matters when printing final products or artistic objects. In addition, this method can also generate a thin and strong structure.

By using the helical/spiral 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

4

support material. As such, this print method allows high-angle overhangs. Conventional 3D printing methods do not allow low-angle overhang, e.g., more than 75 degrees, without support material. Because support material causes several problems such as extra material consumption and difficulty in removing it, it is beneficial if no support is required. Although shapes that can be printed by this method are restricted, it is still beneficial. To print a widening or narrowing helix (i.e., a helix with increasing or decreasing radius) using this method, cross section and printing speed must be properly controlled.

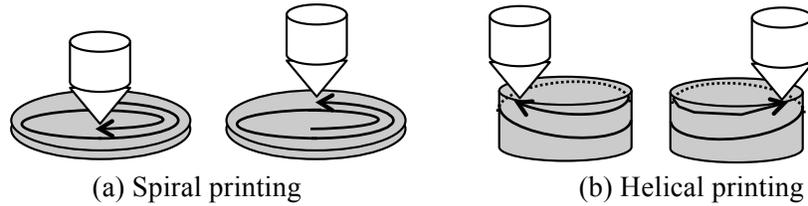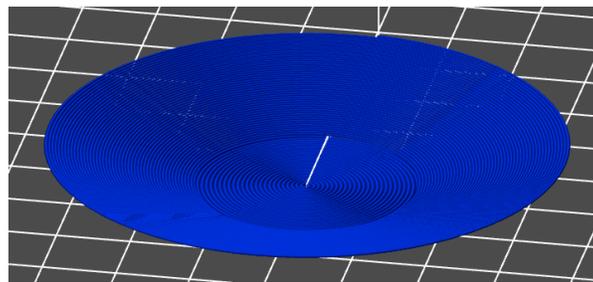(a) Spiral printing   (b) Helical printing

Figure 5: Helical/spiral printing method

Figure 6 shows an example, i.e., a shallow plate. The plate is represented by a G-code program, which is visualized in Figure 6(a). A printed plate is shown in Figure 6(b). No support is required to print it and there are no seams cause by layer transitions. However, if the amount of filament is insufficient or excessive, the printing process easily fails. The problem and a solution is described in Section 4.2.

(a) G-code

(b) Printed plate

Figure 6: G-code and print of shallow and thin plate

3.2 Representation of object models

To print objects by the helical/spiral printing method, the objects must be represented by G-code. However, because G-code is a very low-level (assembly-language level) language, it is not appropriate for handling objects in high-level. An abstract representation called "string" is thus

used for representing objects. Each string Si can be represented by

$$S i = (P i_{\text{start}}, P i_{\text{end}}, c i, v i)$$

where Pi_{start} means the start point of the string, Pi_{end} means the end point of the string, ci means the cross section of the string (which may be replaced by a filament-density parameter), and vi means the printing speed. vi is conceptually unnecessary, but it is useful in implementing (printing) the string. This representation is called a "peeled model" because an object is represented by a sequence of "peels" (i.e., strings). The process that generates strings from an object was called "hashing" in a previous paper [Kan 14a] (and in [Kan 13]).

3.3 Texture-mapping technique

This section summarizes a texture-mapping method (or modulation method) used with the helical/spiral printing method, which is described more in another paper [Kan 15c]. Pictures, characters, or textures can be mapped to the surface of an object printed by helical/spiral printing method by using this method.

To map textures to the surface of a printed object, the cross section of the filament on the surface can be controlled to express textures. This method can generate fine structures, although it is not suited for generating large and deep structures. It can be called the extrusion modulation method; that is, the process of filament extrusion is modulated by pictures, characters, or textures.

Two methods are available for extrusion modulation. The first method is to vary the extrusion speed of filament and the second method is to vary the print-head motion speed. The second method was selected because it is better in response time. In 3D printers, the delay between the motion of the extruder that extrude the filament and the motion of filament at the nozzle, i.e., the tip of the print head is large. It may take several seconds. The response is, thus, slow. Print heads of 3D printers are usually heavy so they have large inertia; however, the response of the print heads are still much better than the filament response.

A method for modulating a surface of a direction-specified model of a 3D-printed object by varying the head motion (and/or the extrusion speed) is described below. This method converts the original strings Si to new strings Si' ($1 \leq i \leq N$). The original and modulated models are represented by sequences of strings described in Section 2. The original model is modulated by using a bitmap and the modulated model is generated.

Figure 7 outlines the modulation process. Each string has the cross section, c_i, and the head speed, v_i, as its properties. These values are updated by the modulation. If the corresponding bit is zero, the value of ci becomes c0 and the value of vi becomes v0. If the corresponding bit is one, the value of ci becomes c1 and the value of vi becomes v1. If v0 = v1, the cross section is controlled only by extrusion speed. This is *not* the selected method. If c1/c0 = v1/v0, the cross section is controlled only by print-head motion speed. This is the selected method. (Note that ci is not the filament extrusion speed that should be specified in the G-code.)
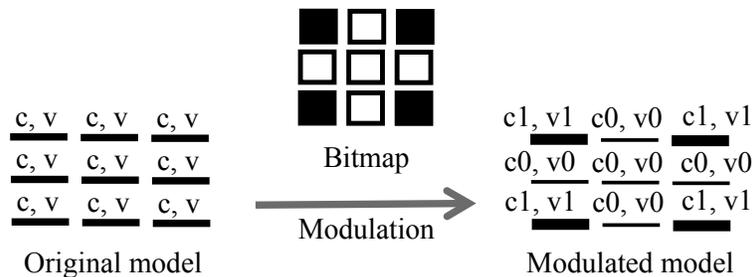

Figure 7: Modulation by a bitmap

Two additional techniques are described here. First, if the bitmap is multi-valued (e.g., the range of the values is 0, 1, 2, 3, or it is continuous), the value of c_i and v_i are also multi-valued. However, the contrast, i.e., the value of c_1/c_0, is restricted to be low because too small or too much extrusion disables stacking filament correctly, so a multi-valued surface structure is not very effective. Second, if the length of a string is too long and the bitmap is very fine, it may have to be divided before mapping the bitmap.

3.4 Light-reflection control technique

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 techniques. This is an attractive attribute for artistic or visual-design purposes. For example, transparent PLA, especially pure PLA, is quite attractive in reflection (see Figure 11); that is, reflection becomes strong in 3D-printed objects made of PLA because strings increase the surface area that reflects light. This attribute is caused not only by the surface, but it is also affected by the internal structure of an object. If the filament is not transparent, the brilliance disappears. The reflection can be controlled by the overhang angle, filament density, or some other attribute of the helical/spiral printing method (**Figure 8**). Figure 8(a) shows reflection controlled by the overhang angle. If the angle of the light is changed, different portions of the object more strongly reflect the light. Figure 8(b) shows reflection controlled by the filament density. Even if the angle of the light is changed, the dark part never reflects light brilliantly.

(a) Reflection controlled by the overhang angle

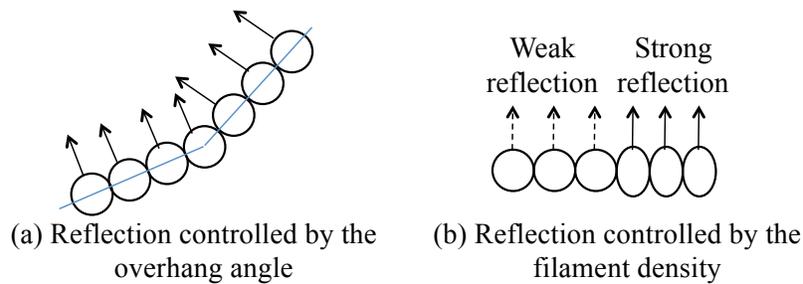(b) Reflection controlled by the filament density

Figure 8: Reflection control

4. Deformation

In this section, first, a problem concerning deformation in 3D printing, which does not exist in graphics, and a solution to the problem are described. Second, the printability preservation concept is described, and third, the results of deformation and other techniques are shown by printed example.

4.1 Deformation method for directed 3D printing

This section describes a method for creating various shapes including complex shapes by using "deformation". This method will be described in detail in another paper [Kan 15a]. To preserve the printability of the object (i.e., to keep the model correctly printable), deformation for 3D printing requires controlling two attributes of strings: cross section and printing velocity. 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 sections, and printing speed.

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

$$\text{deform\_xyz}(\,fd(x, y, z), fc(c, x, y, z), fv(v, x, y, z))$$

In this expression, 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 speed 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 cross section 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 type of mapping can be specified for fd, fv, and k. However, they must at least be continuous and smooth to get a correct printing result.

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

A cylinder coordinate is more useful for describing a deformation, especially when axisymmetric models are deformed.

$$\text{deform\_cylinder}(\,fd(r, \theta, z), fc(c, r, \theta, z), fv(v, r, \theta, z))$$

In this expression, function $fd(r, \theta, z)$ (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 $fc(c, r, \theta, z)$ (i.e., the second argument) maps a cross section at location (r, θ, z). Function $fv(v, r, \theta, z)$ (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, and functions fd, fv, and fc must be continuous and smooth.

Examples of deformation are visualized in **Figure 9**. (Repetier Host was used.) Figure 9(a) shows a cup, which consists of an empty cylinder and a thin cylinder (i.e., the bottom); that is, the cup is an assembly of two direction-specified components. This cup can be transformed to a plate shown in Figure 6 by applying the following deformation to the empty cylinder part.

$$\text{deform\_cylinder}(\,fdd(r, \theta, z), fcd(c, r, \theta, z), fvd(v, r, \theta, z))$$
$$\text{where } fdd(r, \theta, z) = (r + 1.05\,z, \theta, 0.3\,z), \text{ and}$$
$$fcd(c, r, \theta, z) = 0.96\,c, \text{ and } fvd(v, r, \theta, z) = v.$$

The thin cylinder, i.e., the bottom, must be resized to fit the deformed empty cylinder. The above deformation can be applied before or after the assembly. Other examples, i.e., Figure 9(b) and (c) can be created in similar ways.

Figure 9(d) shows an empty cylinder (without a bottom). This cylinder can be transformed to a sphere shown in Figue 9(e), which preserves the filament pitch, by applying the following deformation.

$$\text{deform\_cylinder}(\,fds(r, \theta, z), fcs(c, r, \theta, z), fvs(v, r, \theta, z))$$
$$\text{where } fds(r, \theta, z) = (Radius * \sin(\pi z \,/\, cylinderHeight), \theta,$$
$$r - Radius * \cos(\pi z \,/\, cylinderHeight)),$$
$$fcs(c, r, \theta, z) = 1.2\,c, \text{ and}$$
$$fvs(v, r, \theta, z) = 1.2\,((\,fr(r, \theta, z) \,/\, Radius)^2 + 0.1)\,v$$
$$(\,fr \text{ is the first component of } fds \text{ (i.e., the radius))}$$

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.

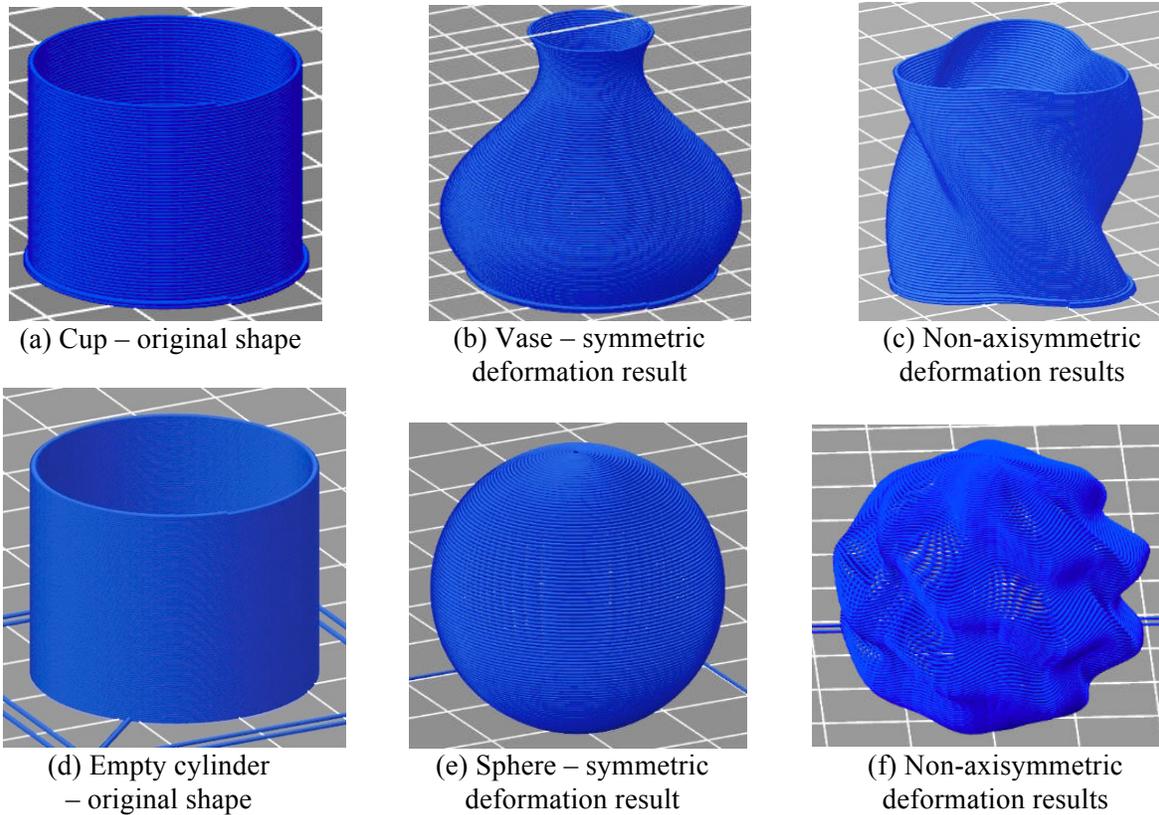| (a) Cup – original shape | (b) Vase – symmetric deformation result | (c) Non-axisymmetric deformation results |
| (d) Empty cylinder – original shape | (e) Sphere – symmetric deformation result | (f) Non-axisymmetric deformation results |

Figure 9: Examples of deformation

Moreover, non-axisymmetric shapes such as those shown in Figure 9(c) or (f), which can be generated by deforming axisymmetric shapes shown in Figures 9(a) and (d), can be easily generated using cylinder-coordinate based deformations. Although various shapes exist in Figure 9, all these shapes are generated only using these trigonometric functions. However, other types of functions can of course be used.

4.2 Printability preservation

The concept of 3D printability preservation plays an important role in avoiding printing failures and in creating exact shapes by the proposed method. If a model is assumed to be printable before a deformation and the deformation preserves the printability, the deformed model is printable. For example, if the deformation reduces the string's density, the cross section must be increased to preserve the printability. If the original shape is simple, it is easier to make it printable. A printable complex shape can be created by deformation with the printability preservation method.

If printability is not preserved, helical/spiral printing may easily fail. **Figure 10** shows two cases of failures. If the amount of extrusion is constant, a new string may fail to be stacked and may drop (Figure 10(a)). On the contrary, if the deformation increases the string density, the cross section must be decreased. Otherwise, excess filament may cause bumps because of the elasticity of printed filament (Figure 10(b)). The filament extrusion speed and also the print-head motion-speed must thus be carefully controlled. The detail is described in another paper [Kan 15a].

If extruded filament becomes solid immediately, there is probably no need to update the printing speed. The printing speed is dependent only on the mechanical features of the 3D printer.

However, because it takes some time for filament to become solid, if the deformation makes the round-trip time of the print head shorter, the printing speed must be reduced to keep the round-trip time longer (it may have to be constant). The *round-trip time* described above means the time required to touch old and new extruded filament. Therefore, deformation updates the printing speed of the model by *fv*.

(a) Dropped sparse filament
(cross section)

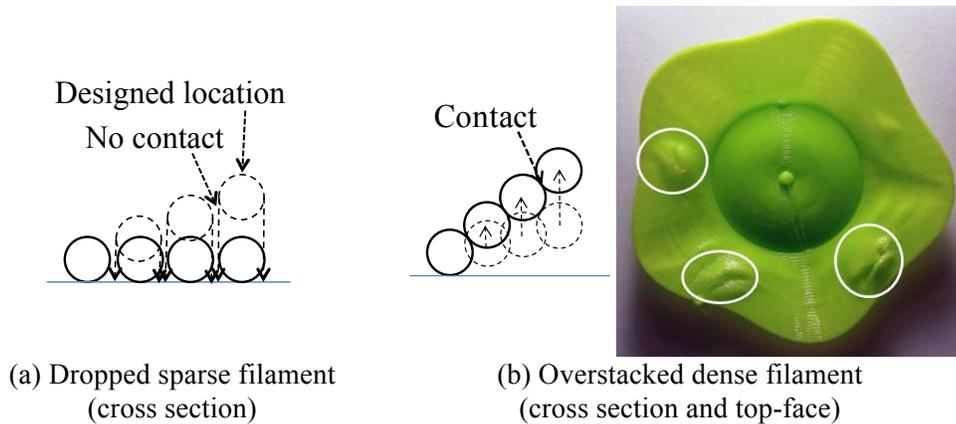(b) Overstacked dense filament
(cross section and top-face)

Figure 10: Printing failures

A peeled model is defined to be *3D printable* (or to have 3D printability) if it can be translated into a numerical-control (NC) program that prints a 3D object of a designed form. If the translated program usually or often fails to generate a designed object, the model is not 3D printable. For example, the coordinate of printed filament may have large error compared to the specified coordinate of the string in the model. This means the model is not 3D printable. This 3D printability concept still has many ambiguities, which must be resolved to make the concept unambiguous.

Original models (shapes) used for this method must be 3D printable, and the modeling and deformation steps for this method must preserve 3D printability. To preserve 3D printability, the cross section and the printing speed specified in the strings in the model must be updated correctly. If a deformation increases or decreases the density of strings, the cross section must also be increased or decreased. If the density of strings is initially $n0$ (/ mm^2) and it becomes $n1$ and the cross section is initially $c0$, the cross section should become $c0 * n0 / n1$ (approximately).

4.3 Printing plates and vases

Various plates and vases can be created by deforming a cup, as shown in Figure 9(a); that is, a combination of an empty cylinder and a thin cylinder. The objects printed by Rostock Max 3D-printer are shown in **Figures 11**(a) to (d). All the samples in this section and in Section 2 are available from a Web site (http://store.shopping.yahoo.co.jp/dasyn/).

The deformation function used for the plate shown in Figure 11(a) contains cos(4θ), which generates the 4-cycle patterns, and the light-reflection control technique[2]. The brighter areas move while changing the light-source direction. Figure 11(b) shows a heart-shaped plate[3]. The deformation function for this plate is based on the following function that maps a circle to a heart shape.

---

[2] A video on the printing process (http://youtu.be/5P1vaahzW98) and samples (http://store.shopping.yahoo.co.jp/dasyn/1011-04.html) are available.
[3] A video and samples: http://youtu.be/G9x14DZYN_8, http://store.shopping.yahoo. co.jp/dasyn/3db0f5bafe.html

$$fdh(x, y, z) = (x + bz \text{ sqrt}(|y| / radius), y, z)$$

This function is based on a "equations for heard-shaped curve" [Yam 07]. The appropriate range of parameter b is from 0 to 1.2. This function becomes identity function when $bz = 0$ (at the bottom of the plate), and it generates sharper heart shape when bz becomes larger (at the top).

(a) 4-cycle plate   (b) Heart-shaped plate   (c) Vertically-"swinging" vase   (d) Wine glass
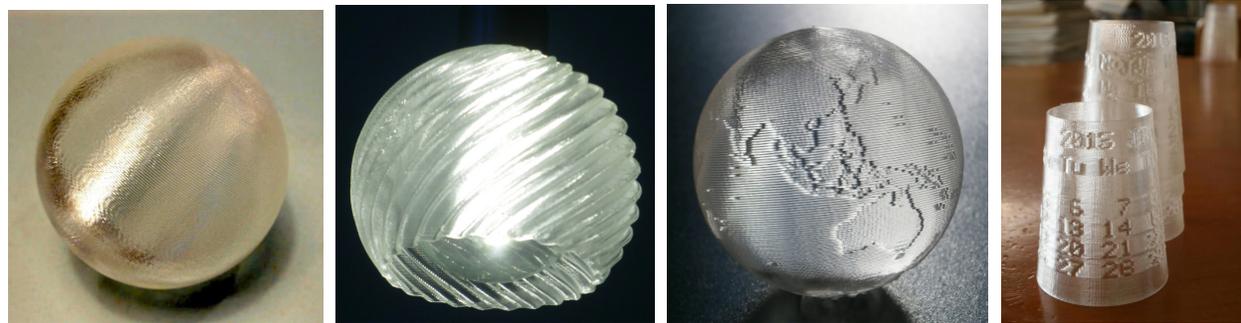
(e) Sphere   (f) LED shade   (g) Globe   (h) Calendar

Fig. 11: Printed objects generated by various deformations

Figure 11(c) shows a vertically "swinging" vase. It was printed with three-cycle vertical motion. Figure 11(d) shows a "wine glass", which is different from the usual wine glass in shape; that is, wine goes into the stem because there is no raised bottom. They are generated from a cup as well. Vases and wine glasses sometimes leak water; however, by controlling the filament cross-section and the printing speed properly, most of them do not leak water. The shapes of all the plates 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 Figure 11(a) to (c). The plates will look better if they consist of finer lines, but it will take more time to print them.

4.4 Printing spheres and globes

A sphere can be created by deforming a thin cylinder, as shown in Figure 9(d). An example of a printed sphere is shown in Figure 11(e)[4]. Figure 11(f) shows a shade for an LED lamp. This is the largest production in Figure 11; however the diameter is approximately 80 mm, and the printing time is less than 20 minutes. Figure 11(g) shows a globe, i.e., a sphere that a world map of 300x150 pixels is mapped to; this means the sphere consists of 150 approximate circles and each circle consists of 300 strings[5]. The map by equidistant cylindrical projection method derived from NASA was taken from a web site called "Celsia Motherload" (http://www.celestiamotherlode.net/catalog/earth.php). The diameter is 50 mm and the contrast of

---

[4] http://youtu.be/xr6zg0Z07HA, http://store.shopping.yahoo.co.jp/dasyn/1032-10.html
[5] http://store.shopping.yahoo.co.jp/dasyn/1201-03.html, http://youtu.be/YWx1vqig2-o

the land and the sea is 1.3 to 1.4. If the contrast is too strong, the printing may fail. Figure 11(h) shows part of a calendar. Each cylinder contains days in two months, so a set of calendar consists of six cylinders.

5. CAD-based Modeling/Manufacturing Method

This section outlines a proposed methodology for designing and printing directed 3D objects, which is a refined version of the methodology proposed in the previous paper [Kan 14a], and outlines an implementation.

5.1 Methodology

The helical/spiral printing method can be used with two types of modeling and manufacturing methods: a 3D-CAD-based method and a generative method. The CAD-based method is similar to conventional CAD-based method. It is used for top-down (artificial and intentional) design. Object models can be created by using manual-design tools or created by assembling predefined parts by using a pointing device such as a mouse or a tablet. The models are represented by a declarative language, such as STL or OpenSCAD [Wik 15]. In contrast, the generative method is used for bottom-up (emergent and non-intentional) design. Generative-design tools generate models algorithmically using a procedural language, such as Processing [Pea 11]. The 3D-printing turtle-graphics tool [Kan 15e] is intended to be a tool of this type. The users of the tools, i.e., the programmers, therefore write programs to generate models.

Although the helical/spiral printing method can be used with a type of 3D CAD as described above, conventional design and tool-path generation tools, which depends on *layers*, cannot be applied to this method Because this printing method does not generate layers and it is intended to control printing directions. Instead, a new methodology explained below [Kan 14a][Kan 15a] is used for direction-specified 3D design and printing.

The design and printing process of the direction-specified design-and-printing method consists of six steps: field-oriented modeling, partitioning, deformation, modulation (or texture mapping), field-based tool-path generation, and non-horizontal 3D printing (**Figure 12**), which is an extended version of the process described in previous papers [Kan 14a][Kan 15a]. This process can contain both bottom-up and top-down processes. The first four steps are the design steps. Some of these steps are briefly reviewed below using **Figure 13**.
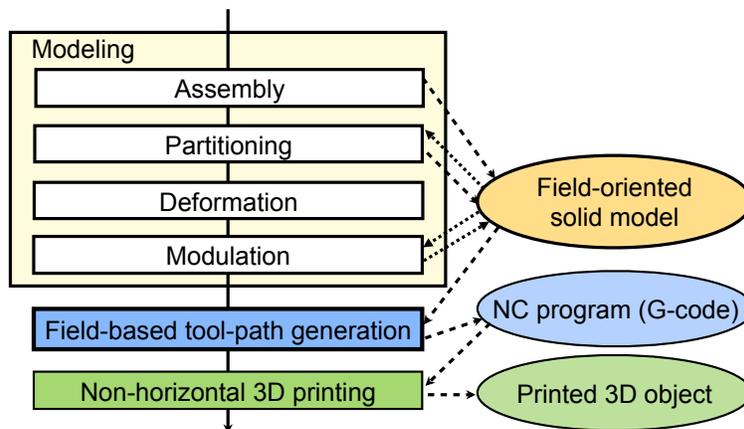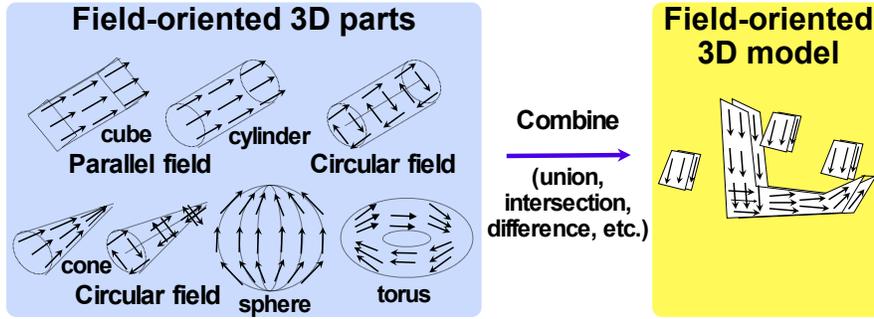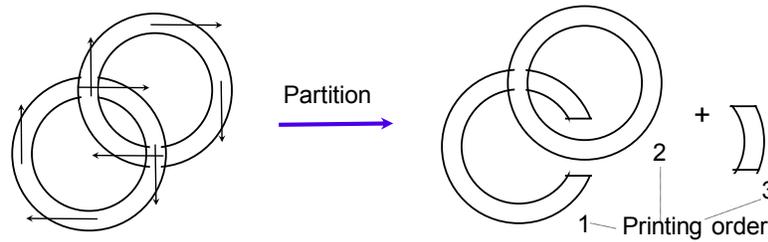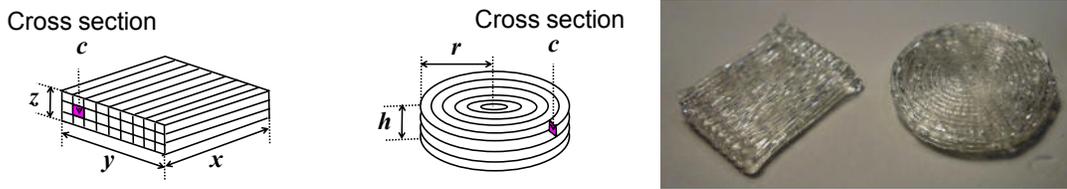

Figire 12: Design and printing process for direction-specified 3D printing

12

(a) Field-oriented modeling by parts combination

(b) Printability enhancement sub-step of the partitioning step

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

Figure 13: Partitioning a model for printability enhancement

In the field-oriented modeling step, an object model with a vector field is created. This model is represented by a modeling language, which can express a vector field. It is an extension of a 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 (Figure 13(a)) [Kan 14a]. The palette in this figure contains various parts and some of them are composite parts.

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 explained in a previous paper [Kan 14a] and thus not explained here, is applied only when the model becomes printable by dividing it into two or more sub-models (see Figure 13(b) for example).

In the peeling sub-step, the model is "peeled" ("hashed") along the field vectors (Figure 13(c)) and a set of *strings* is generated as a result. The peeled-model generation process is completely different from processes used in conventional 3D printing, which is based on the 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 a previous paper [Kan 14a]. In the deformation step, the designed model can be deformed (modulated) to various shapes.

The execution order of the design steps, i.e., the first three steps, is conceptually shown in Figure 12; 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 case, the first step is partitioning, which is included in the part design process, and the second step is deformation.

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 a previous paper [Kan 14a].

Finally, in the non-horizontal 3D printing step, the NC program is executed by a 3D printer and a printed 3D object is outputted.

5.2 Preliminary implementation

The methodology described in the previous subsection has been partially implemented. A preliminary version of a modeler program library, `draw3dp.py` (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, although no GUI-based modeler is available, 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 they will be applied to parts or partially combined model as well. This library was used for printing objects shown in Figure 11.

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 6.1. 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)

The parameters "radius" and "height" determine the shape of the cylinder; the parameters vpitch (i.e., vertical string-pitch) and hpitch (i.e., horizontal string-pitch) determine 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 of the deformations can be described as follows. A Descartes-coordinate based deformation functions to operate directly on these values. A cylinder-coordinate based deformation translates the Descartes coordinates, i.e., x, y, and z, to cylinder coordinates, 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 to reduce numerical errors in a future version.

6. Conclusion

This paper proposes a layer-less FDM method that enables mostly horizontal stacking of filament without support material. Such filament-stacking is enabled by increasing the height of the print head gradually, i.e., without layer transitions that make horizontal stacking difficult. The proposed method also supports techniques for controlling printing directions and various printing-direction-dependent expressions, such as fiber-like textures or brilliance, and "deformation" and "modulation" techniques, which generates various shapes and textures. These techniques make AM products attractive as final products for consumers or as arts. A CAD-based

methodology for modeling and manufacturing direction-specified objects and the helical/spiral printing method is proposed and partially implemented as a Python library. Globes, plates, and other types of then, light objects with brilliant reflections are created by using this methodology and implementation.

References

[Kan 13]   Kanada, Y., "A Method of 3D Printing which is Consistent with Natural Direction in Shape", Poster, *International SFF Symposium 2013*, August 2013. http://www.kanadas.com/papers-e/2013/08/a_method_of_3d_printing_which.html (The content of this paper will be available as [Kan 15b].)

[Kan 14a]   Kanada, Y., "Method of Designing, Partitioning, and Printing 3D Objects with Specified Printing Direction", *2014 International Symposium on Flexible Automation* (*ISFA 2014*).

[Kan 14b]   Kanada, Y., "3D-printing of Generative Art by using Combination and Deformation of Direction-specified 3D Parts," *4th International Conference on Additive Manufacturing and Bio-Manufacturing* (*ICAM-BM 2014, Beijing*), November 2014, http://bit.ly/1zcMCaO

[Kan 15a]   Kanada, Y., "3D Printing of Generative Art using the Assembly and Deformation of Direction-specified Parts", *Rapid Prototyping Journal*, accepted for publication, 2015.

[Kan 15b]   Kanada, Y., "Natural-Direction-Consistent 3D-Design and -Printing Methods", *International Journal of Computer, Control, Quantum and Information Engineering, WASET*, Vol. 9, No. 4, pp. 689-693, 2015.

[Kan 15c]   Kanada, Y., "Creating Thin Objects with Bit-mapped Pictures / Characters by FDM Helical 3D Printing", *8th International Conference on Leading Edge Manufacturing in 21st Century* (*LEM21*), 2015.

[Kan 15d]   Kanada, Y., "3D-printing plates without "support"", to be published, WASET, 2015.

[Kan 15e]   Kanada, Y., ""3D Turtle Graphics" by using a 3D Printer", *Int. Journal of Engineering Research and Applications (IJERA)*, Vol. 5, No 4, Part-5, April 2015, pp.70-77.

[Pea 11]   Pearson, M., "Generative Art: A Practical Guide Using Processing", Manning Publishing Co., 2011.

[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*.

[Wik 15]   WikiBooks, "OpenSCAD User Manual", http://en.wikibooks.org/wiki/OpenSCAD_User_Manual

[Yam 07]   Yamamoto, N., "Heart-shaped Curves", http://www.geocities.jp/nyjp07/heart/index_heart.html (in Japanese).