Designing 3D-Printable Generative Art by 3D Turtle Graphics and Assembly-and-Deformation

Yasusi Kanada
Dasyn.com, Japan
Conventional 3D Printing

► FDM-type 3D printers
- There are many types of 3D printers, but a popular cheap one is called fused deposition modeling (FDM) type.
- They extrude melted plastic from the nozzle and solidify it.
- These printers accept “G-code” (drawing data and program)

Reprap
Stratasys FDM®
Rostock MAX

Two Types of 3D Design Methods for 3D Printing

► Manual design method — conventional method
  ■ 3D CAD/graphics tools and pointing devices are available.
  ■ Examples: Autodesk CAD tools

► Generative design method — proposed method
  ■ Models are algorithmically designed by using design tools
  ■ Examples: Processing (P3D), OpenSCAD
Proposal: Two Methods for Generative 3D Printing

► 1. Turtle-graphics-based Method

► 2. Assembly-and-deformation Method
1. Turtle-graphics-based Method
Turtle Graphics

► Seymore Papert and Logo
  ■ Papert designed Logo programming language for children.
  ■ By using Logo, 2D line art can be drawn by a “turtle” — (2D) turtle graphics

► Drawing commands for turtle graphics
  ■ forward \( d \): move forward by \( d \).
  ■ left \( \theta \): turn left by \( \theta \) (degrees).
  ■ right \( \theta \): turn right by \( \theta \) (degrees).
3D Turtle Graphics

► Drawing commands
  ■ forward $d$, left $\theta$, right $\theta$
  ■ up $\theta$, down $\theta$

► Extended 3D turtle graphics
  ■ Bernd Paysan, “Dragon Graphics”

“Turtle Graphics” by 3D Printing

► Drawing commands are translated into G-code.
  ■ Forward → G1 (moving while printing)

► Turtle coordinates are converted to Descartes coordinates — used in G-code

► Selection of a turtle coordinate system
  ■ Polar coordinates
    ● Coordinates for flight simulators
      — easy to get lost for the direction of gravity
  ■ Cylindrical coordinates — selected
Alternatives and API

► Alternatives

■ Language similar to Logo
■ API (library) for conventional languages

► API for Python, turtle.py, was developed and published.

■ Cylindrical coordinates are used:
  forward(r, z) instead of up/down θ and forward d.
■ Example: helix

```python
turtle.init(FilamentDiameter, HeadTemperature, BedTemperature, DefaultVelocity)
t = turtle.Trace(CrossSection, x0, y0, z0)
dz = 0.4 / 72
for j in range(0, 16):
  for i in range(0, 72):
    t.forward(1, dz)
    t.left(5)
t.draw()
```

■ turtle.py is publicly available.
Development Method

► Following steps are repeated until succeeded.

- **Program description**
  - I use Python and Emacs.

- **Confirmation by graphics**
  - I use a tool called Repetier Host.

- **3D printing**
  - I use Rostock MAX 3D printer.
Printing Process of Skewed Pyramid

youtu.be/7H5-acxQ_RE
Production (Printing Results)

► 2D fractal figure

► Other 2D figure
Production (Printing Results) (cont’d)

_rotation and enlarging/reducing_

- Shrinking patterns
- Expanding pattern
A sparse pattern

- Designed shape

- Printed shape

Some filament dropping is unavoidable.
Problem of Turtle-graphics-based Method

► It is not easy to design printable objects.
  ■ Printed filament must be supported.

► An easier design method is required.
2. Assembly-and-deformation Method
Outline of the Method

► Objects are designed by two steps.

- Python API “draw3dp.py” is being developed.
  - Publicly available.
  - No GUI is available yet.
Helical/spiral Printing Method

► This method is used in the API (draw3dp.py).
► This method prints objects helically or spirally (instead of printing layer-by-layer).

Spiral printing

Helical printing
Deformation

► It is not easy to generate complex shapes only by parts combination.

► “Deformation” generates various shapes and directions in a generative way while preserving printability.

► Original shape and Deformed shapes

helix (empty cylinder)

filled cylinder

deformation

printable!

printable!
Description of Deformations*

► Deformation using Descartes coordinates
    deform_xyz(fd(x, y, z), fc(c, x, y, z), fv(v, x, y, z))
    - **fd**: mapping location (x, y, z) to (x', y', z').
    - **fc**: mapping cross section c at (x, y, z) to c' at (x', y', z').
    - **fv**: mapping printing speed v at (x, y, z) to v' at (x', y', z').

► Deformation using cylinder coordinates
    deform_cylinder(fd(r, θ, z), fc(c, r, θ, z), fv(v, r, θ, z))
    - **fd**: mapping location (r, θ, z), which is expressed in cylinder coordinates, to (r', θ', z').
    - **fc**: mapping cross section c at location (r, θ, z) to c' at (r', θ', z').
    - **fv**: mapping printing speed v at location (r, θ, z) to v' at (r', θ', z').

► Deformations must preserve “printability”.

*Note: The descriptions are simplified for clarity. In actual practice, the deformations would need to be defined more precisely, taking into account the specific requirements and constraints of the printing process.
Deformation: Examples

► Plate:
  deform_cylinder(
      \( fdd(r, \theta, z), fcd(c, r, \theta, z), fvd(v, r, \theta, z) \))
  where \( fdd(r, \theta, z) = (r + 1.05 \, z, \theta, 0.3 \, z) \).

► Vase:
  deform_cylinder(
      \( fdp(r, \theta, z), fcp(c, r, \theta, z), fvp(v, r, \theta, z) \))
  where \( fdp(r, \theta, z) = \\
      (r \, (0.8 + 0.4 \, \sin(z / 8 + 6.5)), \theta, z) \)
Deformation: Examples (cont’d)

► **Sphere:**

deform_cylinder(fds(r, θ, z), fvs(v, r, θ, z), fcs(c, r, θ, z))

where \( fds(r, \theta, z) = \)

\[
(Radius \cdot \sin(z / \text{cylinderHeight}), \theta, r - Radius \cdot \cos(z / \text{cylinderHeight}))
\]
Additional Technique: Texture Mapping

► A method for mapping textures, characters, or pictures on the surface of printed objects is proposed.
  ■ Textures are expressed by difference of cross-sections of filaments.

► Cross-sections of filament fragments are modulated by a bitmap.

Original model

Bitmap

Modulation

Modulated model
Printing Process of Globe

8 times faster. YouTube http://youtu.be/YWx1vqig2-o
Print Results

LED
shade

Globe

Calendar

Sphere

Plates

Deformed sphere

Vase

Globe

Calendar
Thank you