Hairy 3D Printing
Project Report

Nigel Leong, Tan Shun Yu, Zhang Zhexian

15th August, 2017

TABLE OF CONTENTS

1. INTRODUCTION

Aim

In fused deposition modeling (FDM) 3D printing, we generally print objects that have a smooth surface. However, certain features such as lion fur or human hair are difficult to model or fabricate with current 3D printing technology.

Recently, a few 3D modellers have started to experiment with the properties of the plastic filament and parameters of FDM 3D printers to design hairy 3D prints. In the Background section, we will show examples of what we call the “hairy prints”, such as lion fur and horse tail. We learnt that the 3D modelling process of creating the hair is tedious and realised that it can probably be automated by using algorithms.

Our aim is to quicken the hair generation process for 3D printing by automating processes that can be automated such as the generation of the hair fibres in the 3D model. First, the user uploads a 3D object file into a Graphic User Interface (GUI). Next, the user selects regions where he/she wants hair to be generated. After the computation is complete, the user is shown a preview of the new “hairified” 3D model. Finally, users will then download the hairified model which can be printed by a FDM 3D printer. Our solution will take into account the difficulties associated with modelling regions with uneven surfaces, curvature and orientation. We will fabricate several hairy prints for testing.

Technology

Our software solution is an addon to an existing CAD software known as Blender[1]. We created custom a Python script for automated hair generation, and converted the scripts into portable and installable 3rd party plugins.

For hardware prototyping, we test printed several hairy objects using Mankati FDM 3D printers[2], removed the support structure and molded the hair with the heat gun. The final models are shown in the *Results* section.

[1] Blender: https://www.blender.org/
[2] Mankati 3D Printer: http://www.mankati.com/

2. LITERATURE REVIEW

1. Stereolithography (SLA) 3D Hair Printing Program: Cilllia

MIT Media Lab researchers designed a program called Cilllia[3] to generate hairy models that can be printed by SLA 3D Printers. The program provides a user interface that allows the user to choose some parameters relating to the hair's properties such as hair density and hair length. The program does the slicing and model generation on its own without relying on other software such as CAD software or the 3D printer's slicer software. This is done on purpose so that it is able to generate very high density hairy models without crashing the CAD or slicer software. The MIT Media Lab team used a voxel-based model generation method to instruct the printer to print various hair geometry and structure.

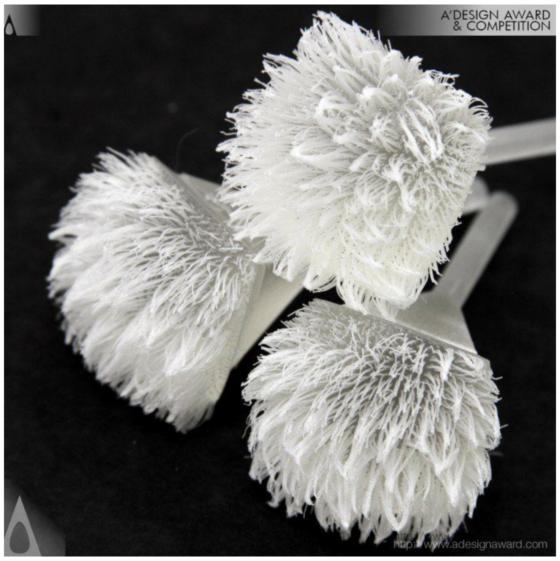


Figure 1. Flowers with hairy texture printed using Cilllia

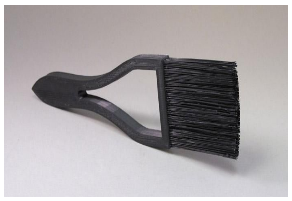
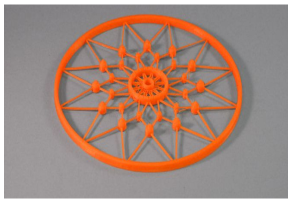
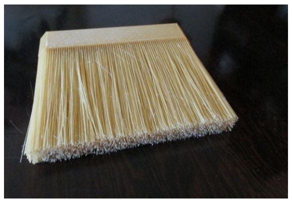
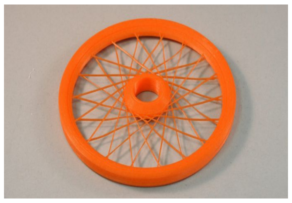
[3] MIT Media Lab Cilllia: https://competition.adesignaward.com/design.php?ID=50708

However, compared to SLA printers, FDM printers are more common and affordable. It may be interesting to print hairy structures with FDM printers.

2. FDM Hairy 3D Prints

Thin hair fibres are able to be printed due to a FDM printing technique called bridging. Bridging refers to the way FDM printers are able to print a plastic fibre using only two supports at the ends of the fibre. This means that there is no need for any supports throughout the length of the plastic fibre. This technique allows us to print interesting objects as shown in the table below.

Mark Leonard, Fibre Bridging Techniques[4]

			
3D Printed Paintbrush	3D Printed Spoked Wheel	3D Printed Broom	3D Printed Dream Catcher

Even without utilising the bridging technique, FDM printers are still able to print strands of plastic fibres. The resulting fibres will curl during the printing process due to the lack of support at the end of the fibre. 3D modellers have played with this effect and came up with a few interesting designs shown below, in Figure 2 to 5.

[4] https://www.spyder3dworld.com/fiber-bridging-techniques-mark-leonard/
https://3dprint.com/32480/3d-print-paintbrush-bridging/

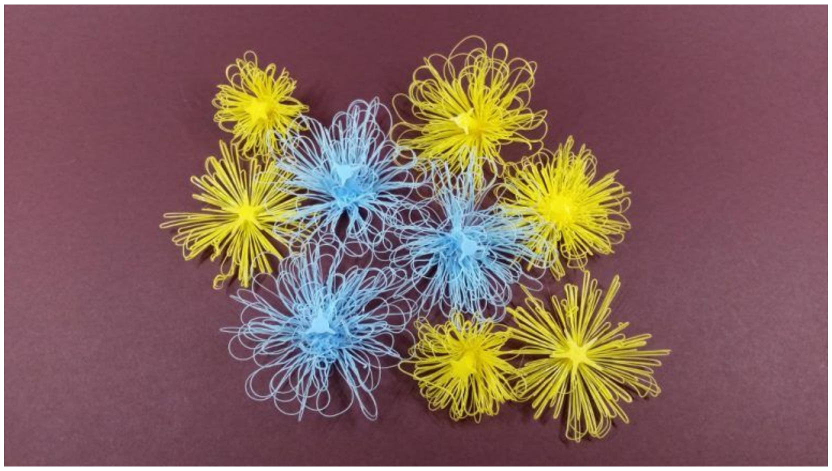


Figure 2. Mark Peeters, the creator of Drooloop Flowers, 2014[5]

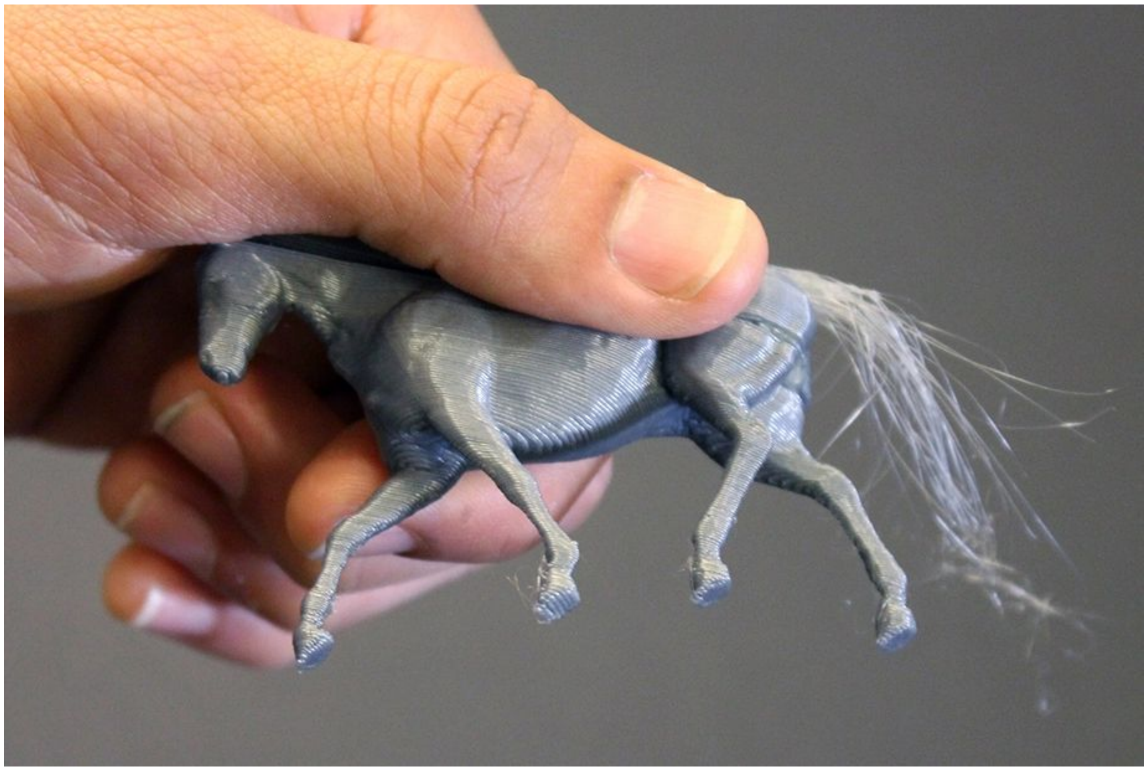


Figure 3. Plastic Horse Tail Hair, 2015[6]

[5] https://www.thingiverse.com/thing:240158
http://3dwithus.com/hairy-3d-prints

[6] https://techcrunch.com/2015/11/04/researchers-can-now-created-3d-printed-plastic-hair/

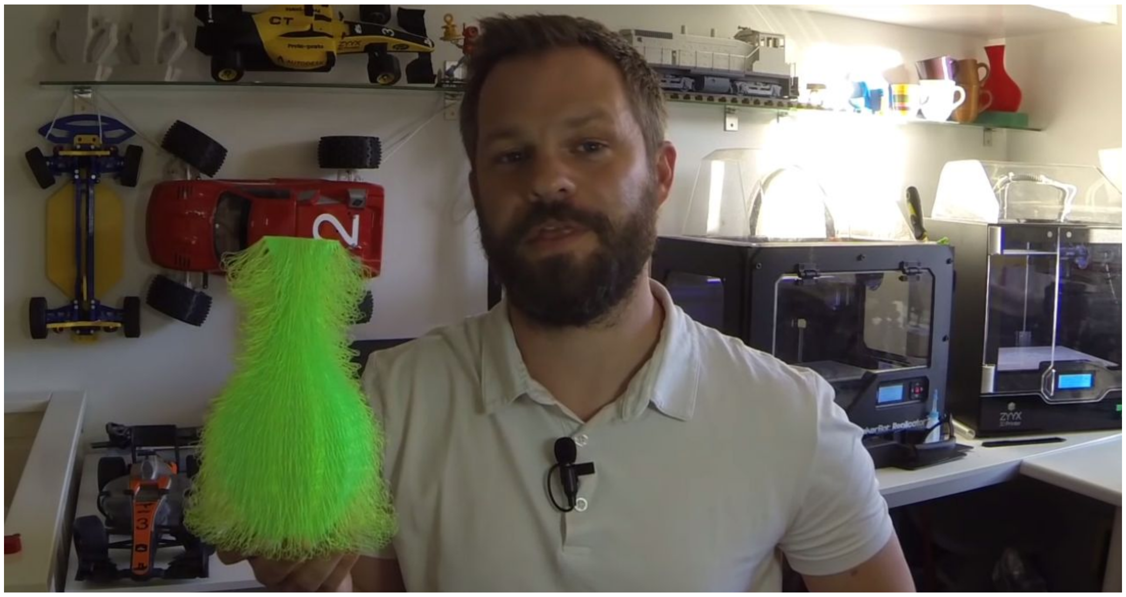


Figure 4. Daniel Norée, Furry Vase, 2016[7]

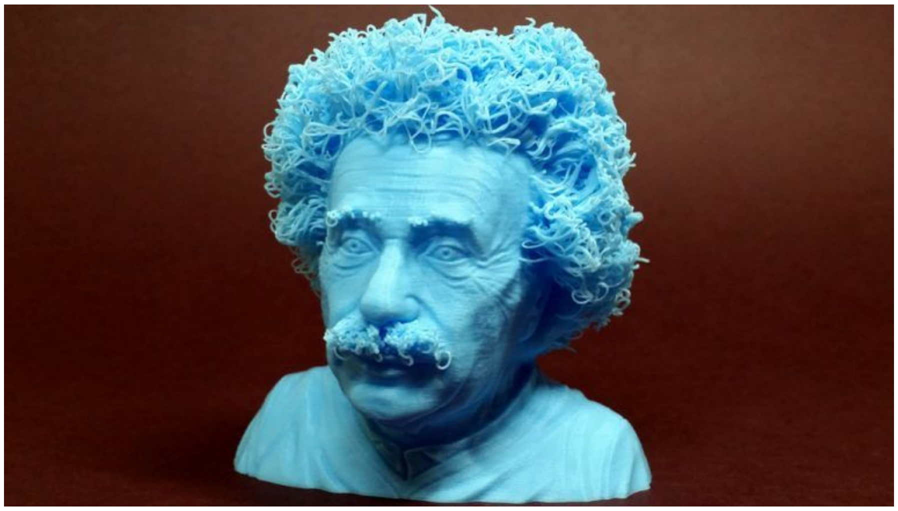


Figure 5. Hairy Einstein, 2017[8]

Hairy Lion, by Primoz Cepin, is one of the hairy 3D prints that became a huge hit. This print uses a cylindrical sacrificial wall to allow the hair fibres to bridge from the lion's face to the wall. This is the print that inspired us to embark on this project. The creator of this print stated that he will not be producing any more hairy 3D objects because it takes too much time to create the hair fibres using Solidworks. Our project hopes to address this problem by automating the generation of the hair fibres on the 3D model.

[7] http://danielnoree.com/?p=786
[8] http://3dwithus.com/hairy-3d-prints

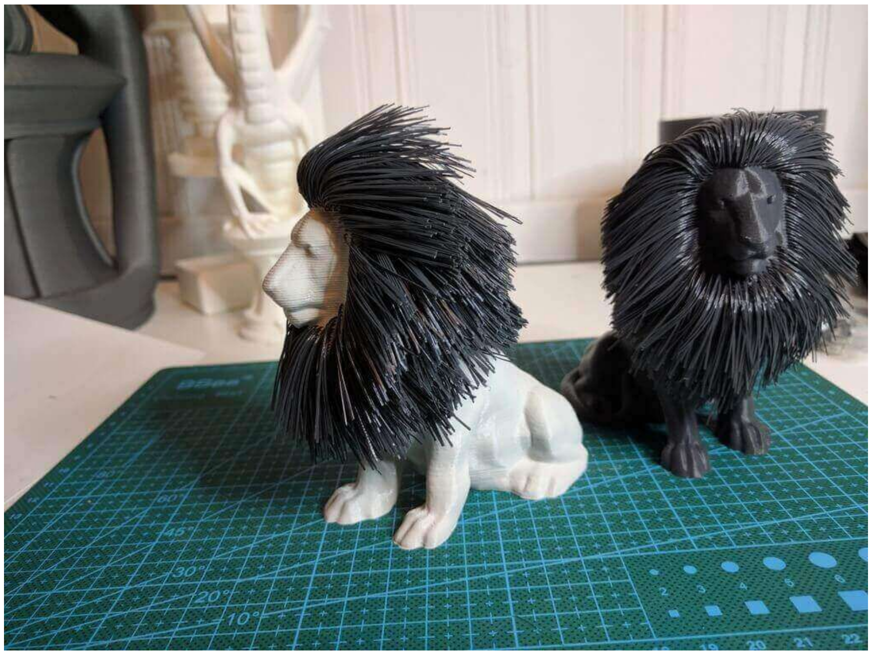


Figure 6. Hairy Lion, by Primoz Cepin[9]

[9] https://all3dp.com/hairy-lion-3d-printing/

3. DEVELOPMENT PROCESS

Overall Workflow

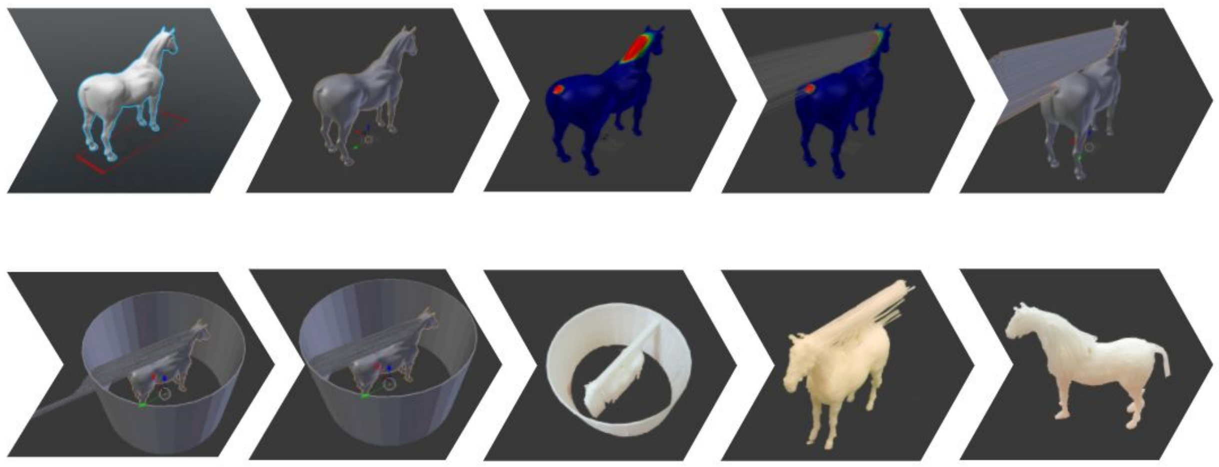


Figure 7. Workflow (top: steps 1-5; botton: steps 6-10)

As shown in Figure 7, the workflow for the project is as follows:

1. Prepare the original .stl file
2. Import .stl file to Blender
3. Select regions to add hair
4. Generate hair using particle effect
5. Convert hair to printable mesh
6. Generate support cylinder (optional)
7. Clean extra hair (if any)
8. 3D printing
9. Remove support structure (if any)

10. Hair shaping

Programming Scripts

1. Hair Generation

- Hair particle effect
- Use modifier to convert to mesh
- Customisable length, thickness and number of hairs

```
bpy.ops.object.particle_system_add()
bpy.data.particles["ParticleSettings"].type = 'HAIR'
bpy.data.particles["ParticleSettings"].count = 3000
bpy.data.particles["ParticleSettings"].hair_length = object_radius
bpy.ops.object.modifier_convert(modifier="ParticleSystem 1")
bpy.ops.object.convert(target='CURVE')
bpy.context.object.data.fill_mode = 'FULL'
bpy.context.object.data.bevel_depth = 0.3
bpy.ops.object.convert(target='MESH')
```

2. Support Cylinder Generation

- Inspired from the hairy lion
- Used for hair bridging
- Dynamic based on object position and size (using bounding box method)

```
# Get parameters of object
object = bpy.context.object
object_location = object.location
object_radius = 0.5 * max(object.bound_box.data.dimensions.x,
object.bound_box.data.dimensions.y)
object_height = object.bound_box.data.dimensions.z

# Parameters
cylinder_radius = object_radius * 1.5 #50
cylinder_thickness = 1
```

```
cylinder_height = object_height #100
cylinder_position = object_location #(0, 0, 0)


# Create an outer cylinder.
bpy.ops.mesh.primitive_cylinder_add(radius=cylinder_radius+cylinder_thickness,depth=cylinder_h
eight)
cylinder_outer = bpy.context.object
cylinder_outer.name = 'cylinder_outer'
cylinder_outer.location = cylinder_position

# Create an inner cylinder.
bpy.ops.mesh.primitive_cylinder_add(radius=cylinder_radius,depth=cylinder_height+20)
cylinder_inner = bpy.context.object
cylinder_inner.name = 'cylinder_inner'
cylinder_inner.location = cylinder_position

# Create a boolean modifier named 'my_bool_mod' for the cube.
mod_bool = cylinder_outer.modifiers.new('my_bool_mod', 'BOOLEAN')
# Set the mode of the modifier to DIFFERENCE.
mod_bool.operation = 'DIFFERENCE'
# Set the object to be used by the modifier.
mod_bool.object = cylinder_inner

# The modifier_apply function only works on the active object.
bpy.context.scene.objects.active = cylinder_outer

# Apply the modifier.
res = bpy.ops.object.modifier_apply(modifier = 'my_bool_mod')

# Delete the cylinder.
cylinder_inner.select = True
bpy.ops.object.delete()
```

3. Mesh Cleanup

There will be hair fibres generated in unwanted locations. This code snippet will clean unwanted hair fibres automatically.

```
### Generate a bigger cylinder to clean the outer hairs ###
# Create a cylinder_cleaning.
bpy.ops.mesh.primitive_cylinder_add(radius=cylinder_radius+cylinder_thickness,depth=cylinder_h
eight)
cylinder_cleaning = bpy.context.object
cylinder_cleaning.name = 'cylinder_cleaning'
cylinder_cleaning.location = cylinder_position

# Create outer cleaning_cylinder.
bpy.ops.mesh.primitive_cylinder_add(radius=cylinder_radius+object_radius+50,depth=cylinder_hei
ght+object_radius+50)
cylinder_outer_outer = bpy.context.object
cylinder_outer_outer.name = 'cylinder_outer_outer'
cylinder_outer_outer.location = cylinder_position
# Create a boolean modifier
mod_bool = cylinder_outer_outer.modifiers.new('my_bool_cleaning', 'BOOLEAN')
# Set the mode of the modifier to DIFFERENCE.
mod_bool.operation = 'DIFFERENCE'
# Set the object to be used by the modifier.
mod_bool.object = cylinder_cleaning
# The modifier_apply function only works on the active object.
bpy.context.scene.objects.active = cylinder_outer_outer
# Apply the modifier.
res = bpy.ops.object.modifier_apply(modifier = 'my_bool_cleaning')

### Clean the hair ###
# Assume that the hair is named 'Mesh'
# Create a boolean modifier
mod_bool = bpy.data.objects['Mesh'].modifiers.new('my_bool_cleaning_hair', 'BOOLEAN')
# Set the mode of the modifier to DIFFERENCE.
mod_bool.operation = 'DIFFERENCE'
# Set the object to be used by the modifier.
mod_bool.object = cylinder_outer_outer
# The modifier_apply function only works on the active object.
```

```
bpy.context.scene.objects.active = bpy.data.objects['Mesh']
# Apply the modifier.
res = bpy.ops.object.modifier_apply(modifier = 'my_bool_cleaning_hair')

# Delete the cylinders.
cylinder_cleaning.select = True
bpy.ops.object.delete()
cylinder_outer_outer.select = True
bpy.ops.object.delete()
```

4. Create Blender Addons

Addon metadata (bl_info variable)

```
bl_info = {
    "name": "Hairy 3D Print",
    "description": "Hairify stl file that can be printed by FDM 3D printers.",
    "author": "Shun Yu, Nigel, Zhexian",
    "version": (1, 0),
    "blender": (2, 78, 0),
    "location": "View3D > UI panel > Add hair",
    "support": "COMMUNITY",
    "category": "Add Mesh"
    }
```

Convert script into class

```
class AutomaticMode(bpy.types.Operator):
    bl_idname = "object.hairfy_auto"
    bl_label = "Hairy 3D Print (Automatic Mode)"
    bl_options = {'REGISTER', 'UNDO'}  # enable undo for the operator.

    def execute(self, context):        # execute() is called by blender when running the
operator.


        return {'FINISHED'}            # this lets blender know the operator finished
successfully.
```

Register and unregister classes

```
def menu_func(self, context):
    self.layout.operator(AutomaticMode.bl_idname)
    self.layout.operator(ManualMode.bl_idname)
    self.layout.operator(ManualMode_Support.bl_idname)

def register():
    bpy.utils.register_module(__name__)
    bpy.types.VIEW3D_MT_object.append(menu_func)

def unregister():
    bpy.utils.unregister_module(__name__)

if __name__ == "__main__":
    register()
```

Post-Printing Processing

The 3D printed hairy structure is likely to have a surrounding support which need to be removed. It can be done by cutting the hair with slim and sharp scissors or penknife, as close to the wall as possible (Figure 8).

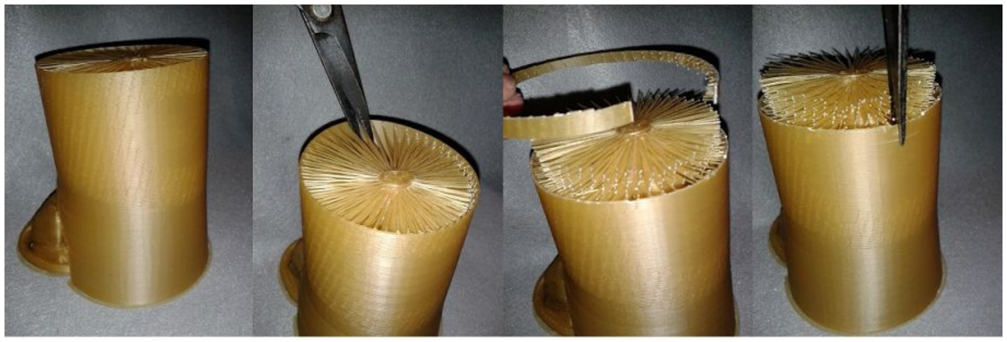


Figure 8. Cutting support structure[10]

As seen in Figure 9, the hair cut from the support structure are unidirectional and unrealistic. To shape the hair, a heat gun may be used to melt the hair, and we may shape the hair by hand. Extra caution needs to be taken when handling the heat gun.

[10] Image source: https://www.thingiverse.com/thing:2007221

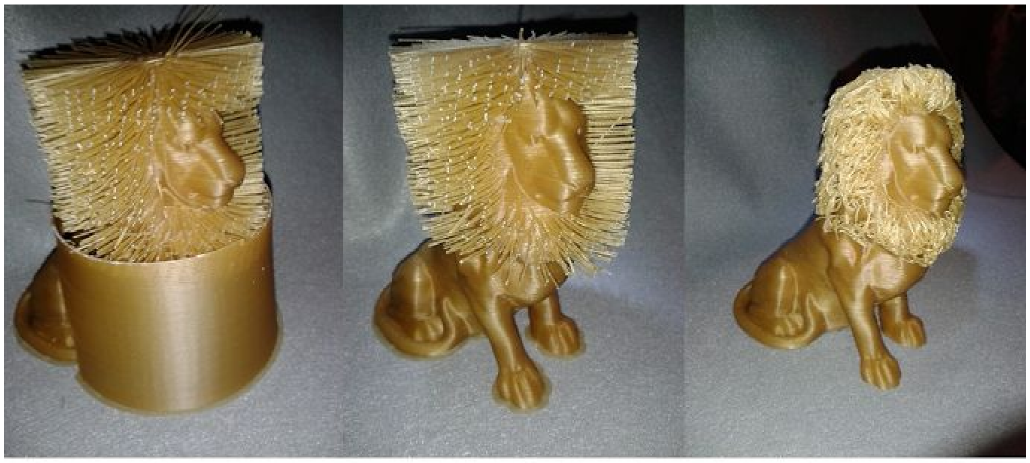


Figure 9. Hair shaping before and after[11]

Challenges & Solutions

Problem Faced	**Solution**
Determining support structure thickness	Trial and error - tried different thicknesses, slice in Mankati and check if cylinder is detected correctly
Determining hair fibre thickness	Trial and error - tried different thicknesses, slice in Mankati and check if hair fibres are detected correctly
Separating individual hair fibres in slicing software	Found an optimal hair fibre thickness that will prevent hair fibres from joining together with other hair fibres
Printing hair near sharp corners	Regard the action of hair generation near object corners as invalid to the user, the user can choose to generate hair near those

[11] Image source: https://www.thingiverse.com/thing:2007221

	areas and comb the 3D printed hair towards those areas
Removing 3D print from printer bed	Optimally aligned cylinder base with object base

4. FINAL PRODUCT

Summary of Prototypes

- **Automatic mode**
    - Cat
    - Bunny
- **Manual mode with cylinder**
    - Christmas Tree
    - Horse
- **Manual mode without cylinder**
    - Christmas Tree
    - Bust (3 different hair styles)

Photos of Prototypes

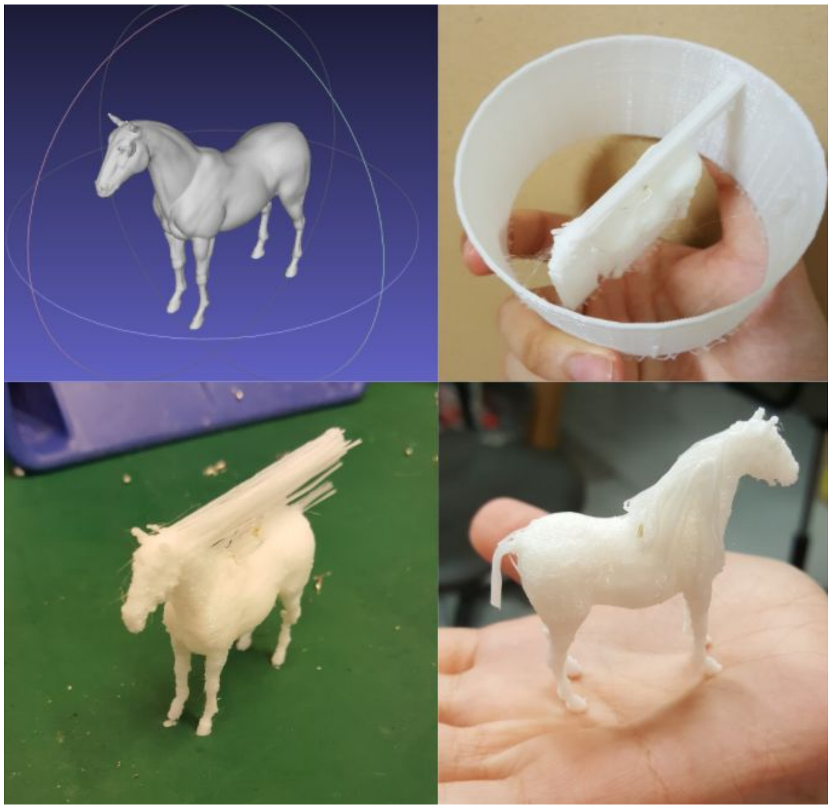


Figure 10. Horse (manual mode with support)

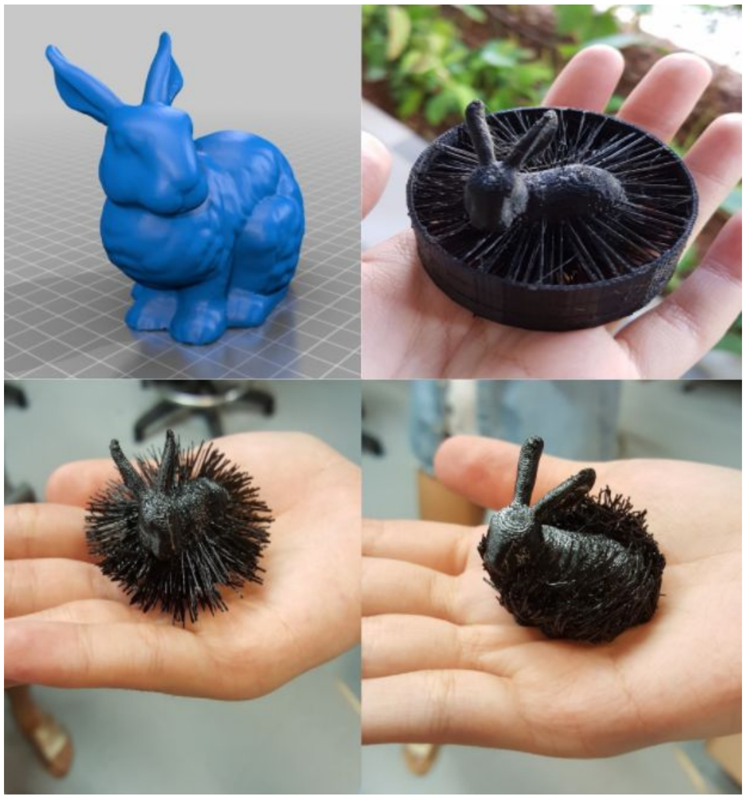


Figure 11. Bunny (automatic mode with support)

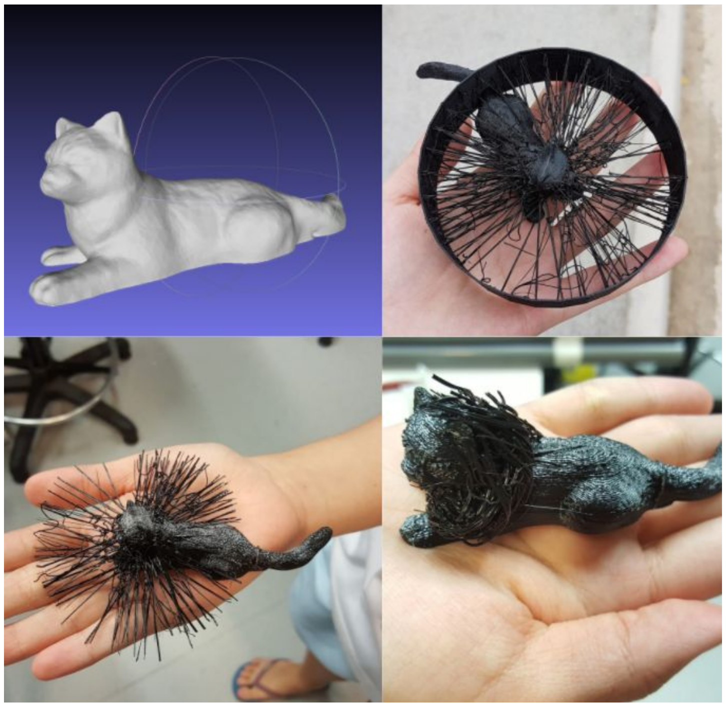


Figure 12. Cat (automatic mode with support)

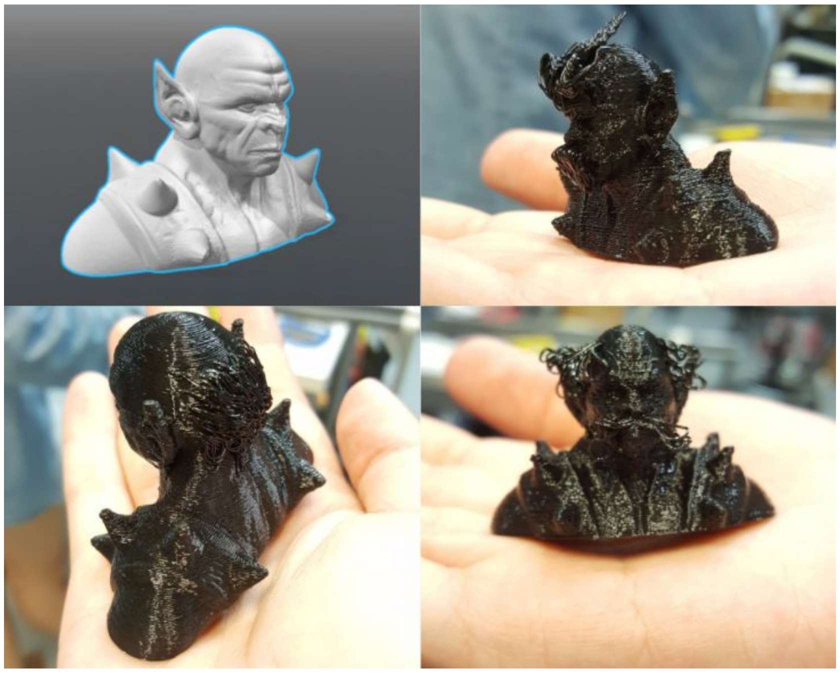


Figure 13. bust (manual mode without support)

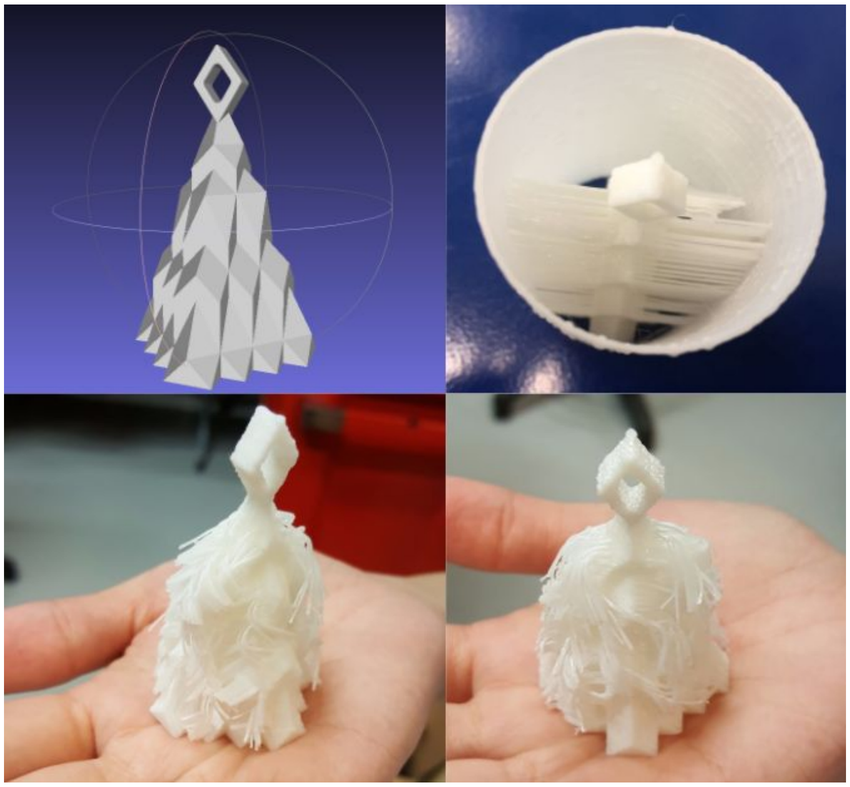


Figure 14. Christmas tree (manual mode with support)

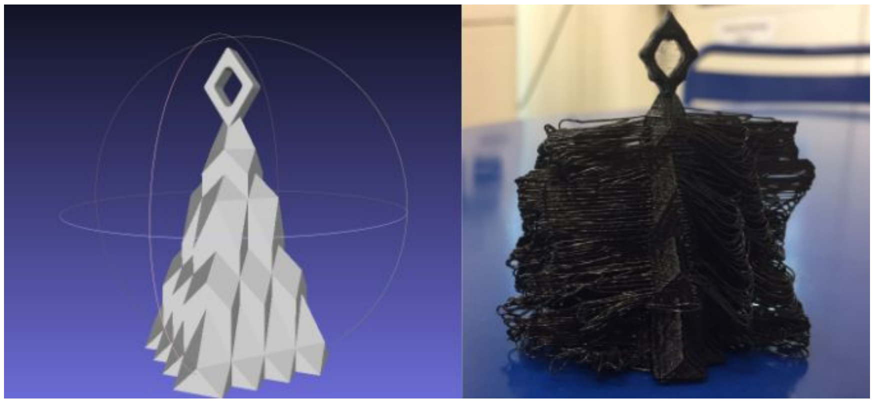


Figure 54. Christmas tree (manual mode without support)

5. FUTURE DEVELOPMENT

Instead of generating a full cylinder around the object as support, we can improve on our cylinder-generation algorithm to generate only a partial portion of support around the area of the object to be hairy. This will not only save on the amount of filament used, but also makes it easier to remove the print from the 3D printer bed and reduces print time.

Currently, to change the parameters of the hair fibres such as hair length, hair thickness and hair density, we have to manually edit the script. To reduce the time and complexity of modifying the parameters, we can develop the Blender addon further to allow the configuration of parameters within Blender's graphical user interface.

As it is physically impossible to allow generation of hairs on surfaces that are perpendicular to the printer bed, we could code some form of indication (i.e. notification) on the Blender add-on advising the user to adjust the printing orientation in order to allow filament to extrude from those surfaces.

REFERENCES

1) Hairy Lion, Thingiverse, https://www.thingiverse.com/thing:2007221
2) FDM Fibre Bridging Techniques, https://www.spyder3dworld.com/fiber-bridging-techniques-mark-leonard/
3) FDM Fibre Bridging 3D Model, https://3dprint.com/32480/3d-print-paintbrush-bridging/
4) Examples of hairy 3D Prints without sacrificial wall, http://3dwithus.com/hairy-3d-prints/
5) Carnegie Mellon hair printing techniques, https://techcrunch.com/2015/11/04/researchers-can-now-created-3d-printed-plastic-hair/
6) Furry Vase, http://danielnoree.com/?p=786
7) Hairy Lion, https://all3dp.com/hairy-lion-3d-printing/
8) Blender - addon tutorial, https://docs.blender.org/api/blender_python_api_2_65_5/info_tutorial_addon.html
9) Blender - Scripting, https://blender.stackexchange.com/questions/65129/how-do-i-create-a-script-for-geometry-i-create/65130
10) SLA Cilllia, https://competition.adesignaward.com/design.php?ID=50708