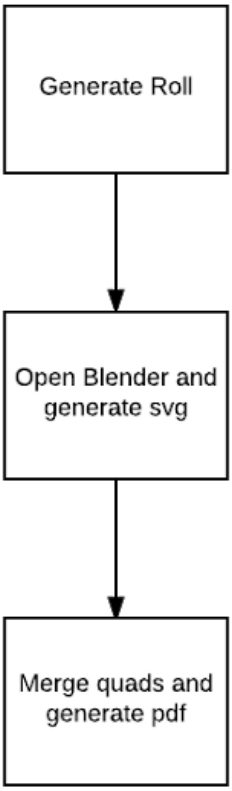
3D Rolling

Software Pipeline



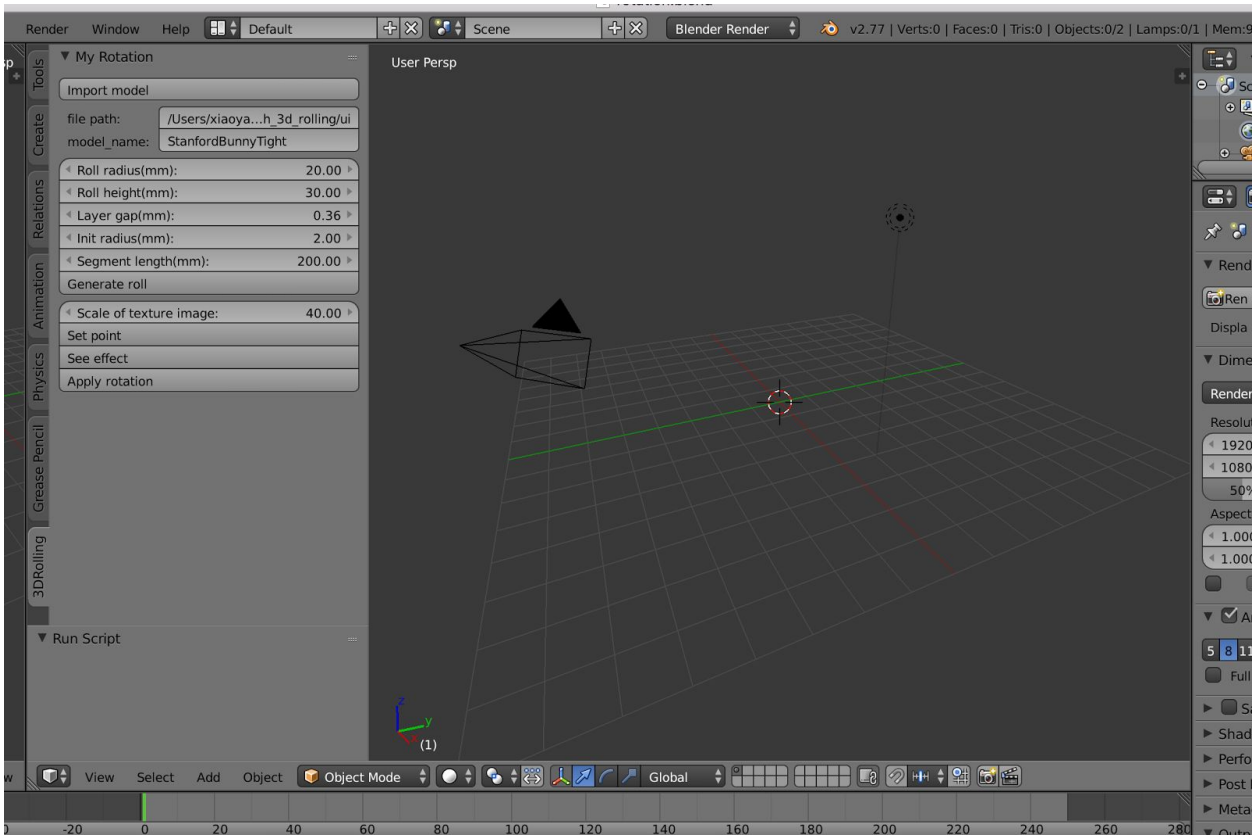
Xiaoyan Wu

INTRODUCTION

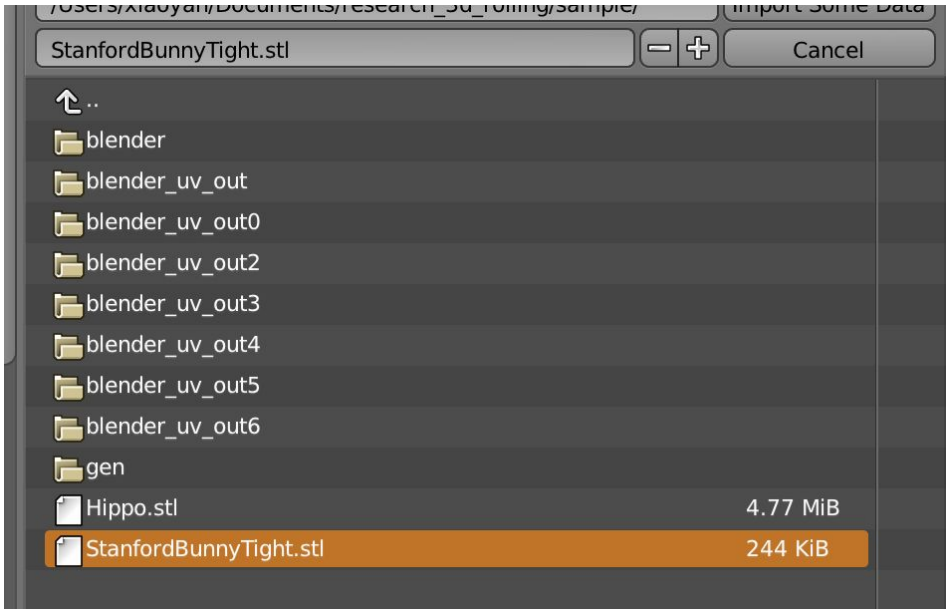
This software pipeline generates the pdf files that can be sent to the laser cutter.

PROCEDURE

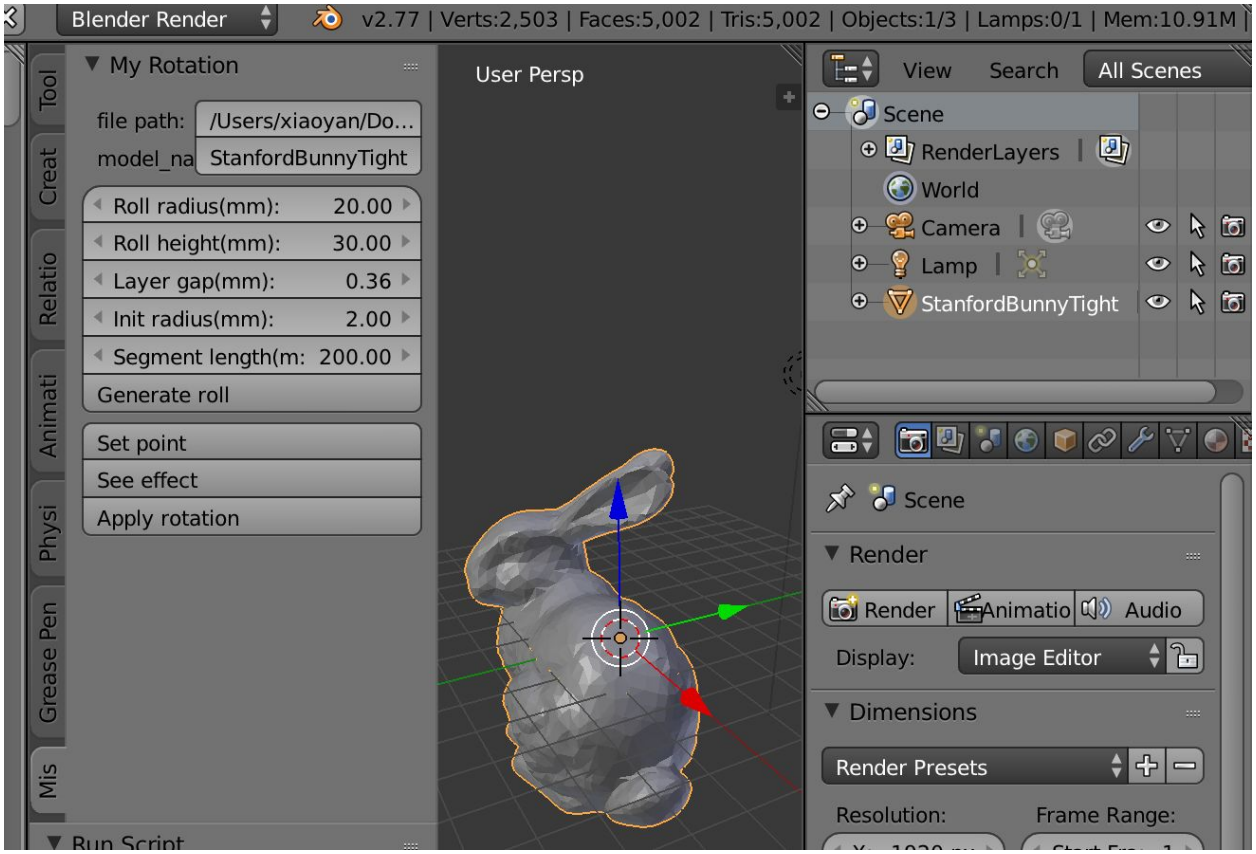
1. Open rotation.blend and you will see an empty blender project with a panel called “3DRolling” as the following screenshot.



Click on the first button “Import model” and it will open a file selector where you can select the stl model you want to use.



The stl model will be imported into blender and there will be a new panel “Misc”. This panel contains three parts. Note that the panel only appears in Object mode.



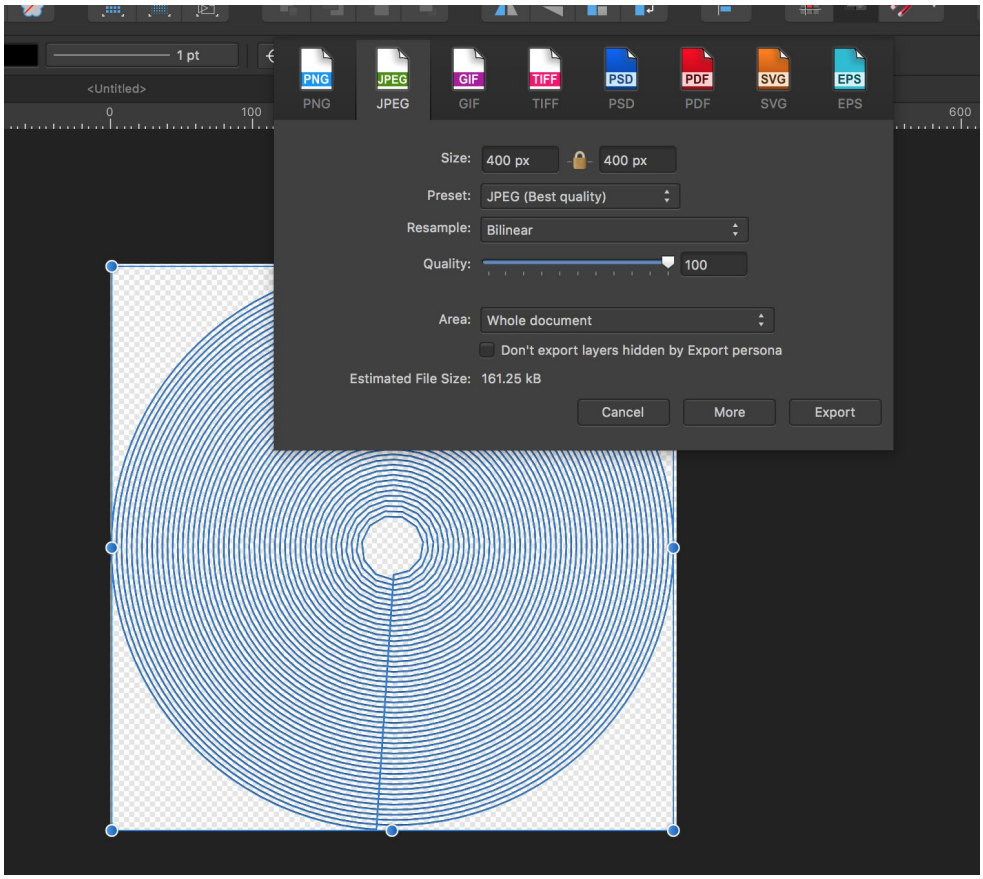
The first part contains two text boxes. The first one is the absolute file path to the folder where the model is placed. The second one is the name of the model imported. It should be the same as the name appeared in the right panel in Blender.

The second part contains five input parameters and a button. The five parameters are as followings. After these 5 parameters are set correctly, click the button “Generate roll” to generate the roll segments. The obj files are placed in a newly created “gen” folder inside the main folder.

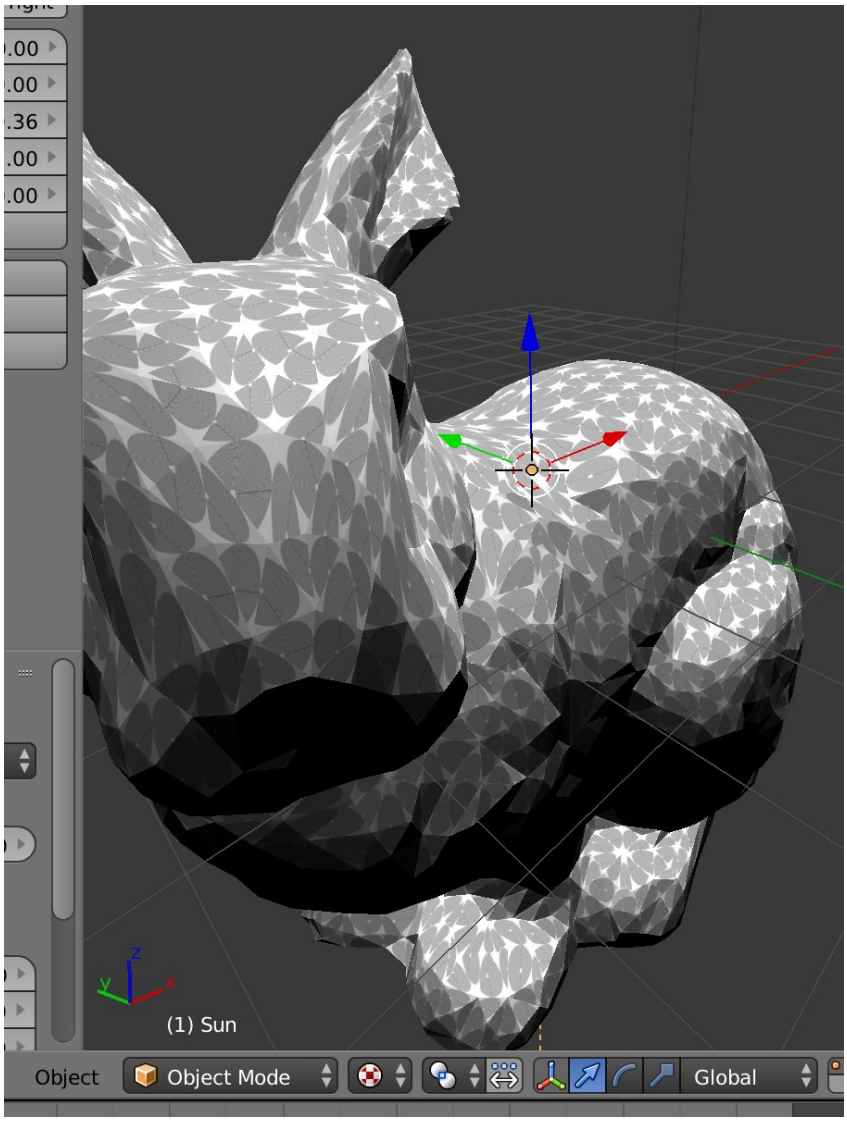
parameters	unit	meaning
Roll radius	mm	Radius of the finished roll
Layer gap	mm	Gap between each layer
Init radius	mm	The radius of the first segment (radius of the rod)

Roll height	mm	Height of the finished roll
Segment length	mm	Length of one segment

While generating the obj files, the code also generates a “curve_test.svg” file that is the cross section of the roll. Open the svg file in Affinity Designer or other software that could open svg files and export the file as jpg. Name it “texture.jpg” and place in the same folder.



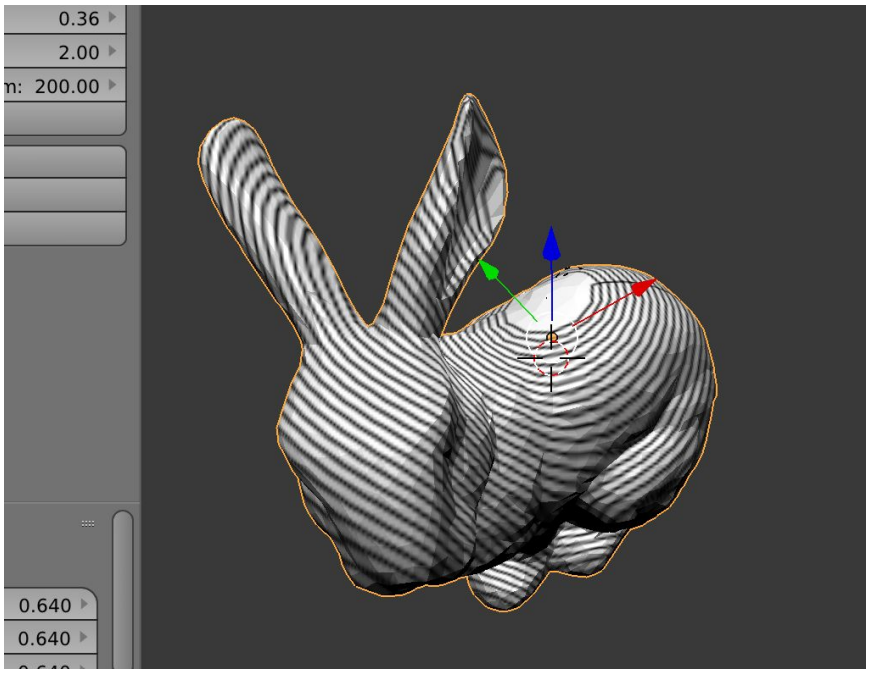
Once there is a “texture.jpg” in the folder, this image will be applied as the texture image for the model. This step is done after generating the roll. At this point, the uv is the default uv unwrapped, and it will change once we set the axis of the roll in the following steps.



The third part contains three buttons, “Set point”, “See effect” and “Apply rotation”. This step is for rotating the roll by setting two points so that the line through these two points becomes the new centerline of the roll.

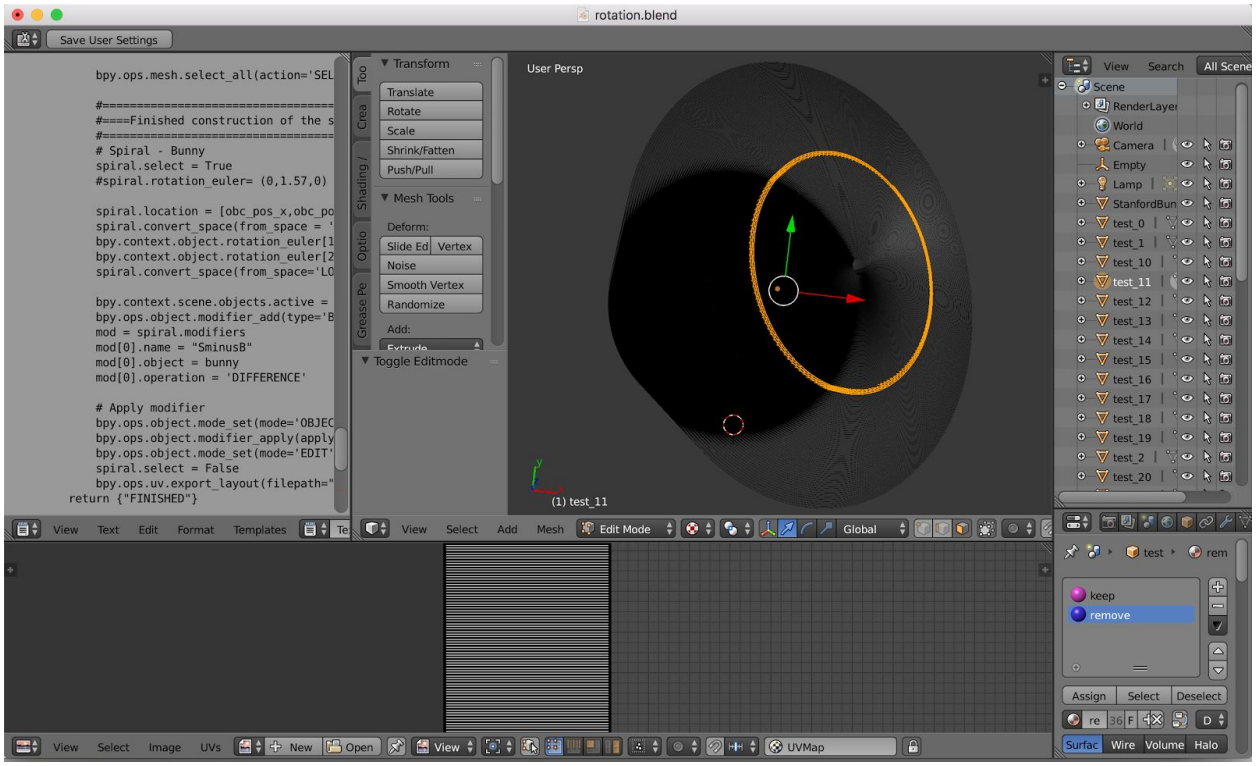
Click on one point on the bunny and click the button “Set point”. This step sets one point in the scene. Since we set “Cursor_depth” to be true, the point will always be on the surface of the bunny.

Then click on another point on the bunny and click “See effect”. This step sets the second point and renders the texture according to these two points. The new texture shows how the cut will be applied. This steps takes less than 1 sec.



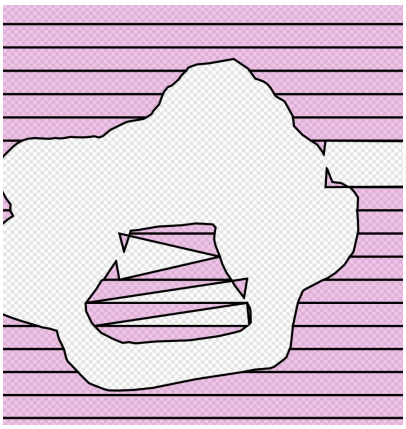
Repeat the first two steps until we find a position that cuts the bunny best.

Then click “Apply rotation” and this step imports the roll segments and apply the cut. It also generates the final pdfs and output them to “blender_uv_out6” folder. This step takes about 10-30 seconds.

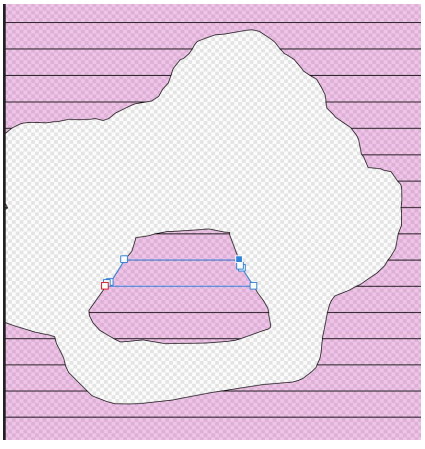


This step is actually a combination of running several python scripts and c scripts that merge the quads in the svg and generate pdf.

The first script is modify.py. Sometimes the svg output from blender does not have all points in the right position. This code puts the wrong points into the right position. This behavior from blender does not always happen, but in case it does, this code fix this issue. The output svg files are placed in “blender_uv_out0” folder.



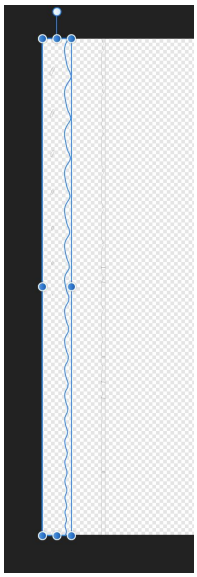
The script is



second parser.py. This code generates the raw files that are needed by the c code. The output is placed in “blender_uv_out2” folder.

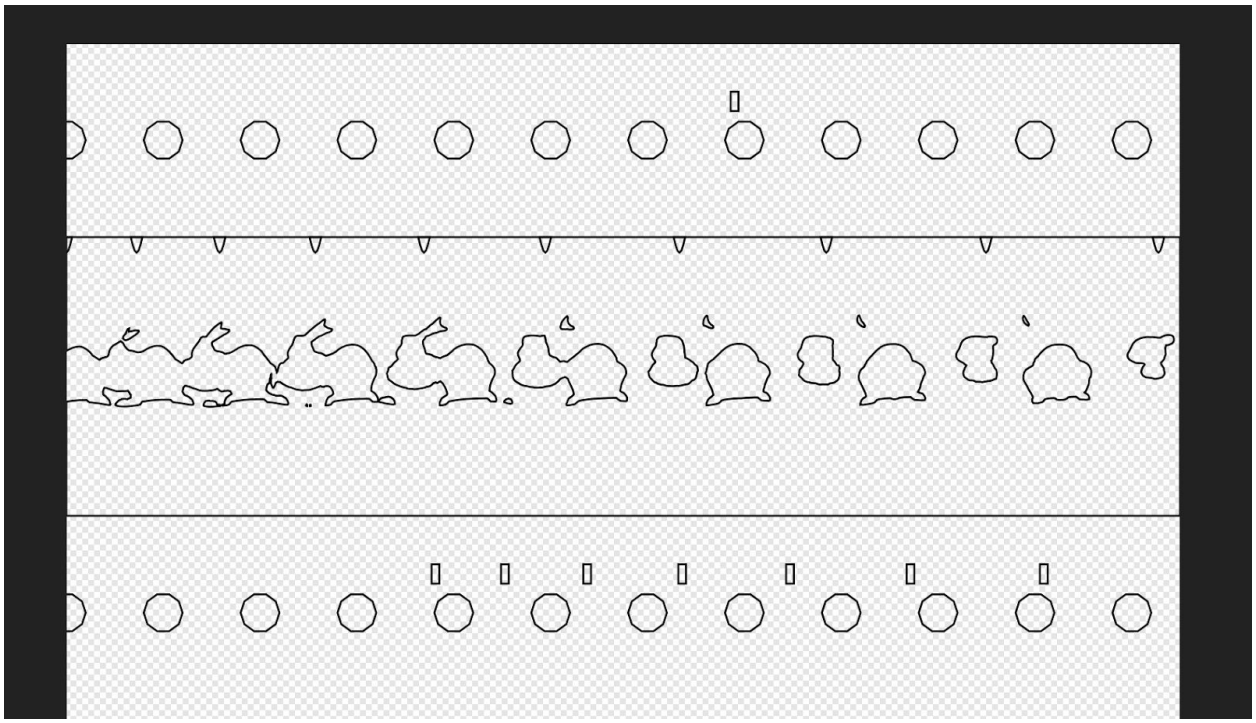
The third script is main.c. It first calls “Make” to update the program and then calls “./exe” to run the program. This is the main code that merge the quads in svg files. It uses a third party library “GPC General Polygon Clipper Library”. The output files are placed in “blender_uv_3” folder.

The fourth script is parser2.py. It converts the c script output to svg files. At this point, the svgs files do not contain any quads. The svg files are placed in “blender_uv_out4” folder. As we can see in the svg below, the quads have been combined into one curve.

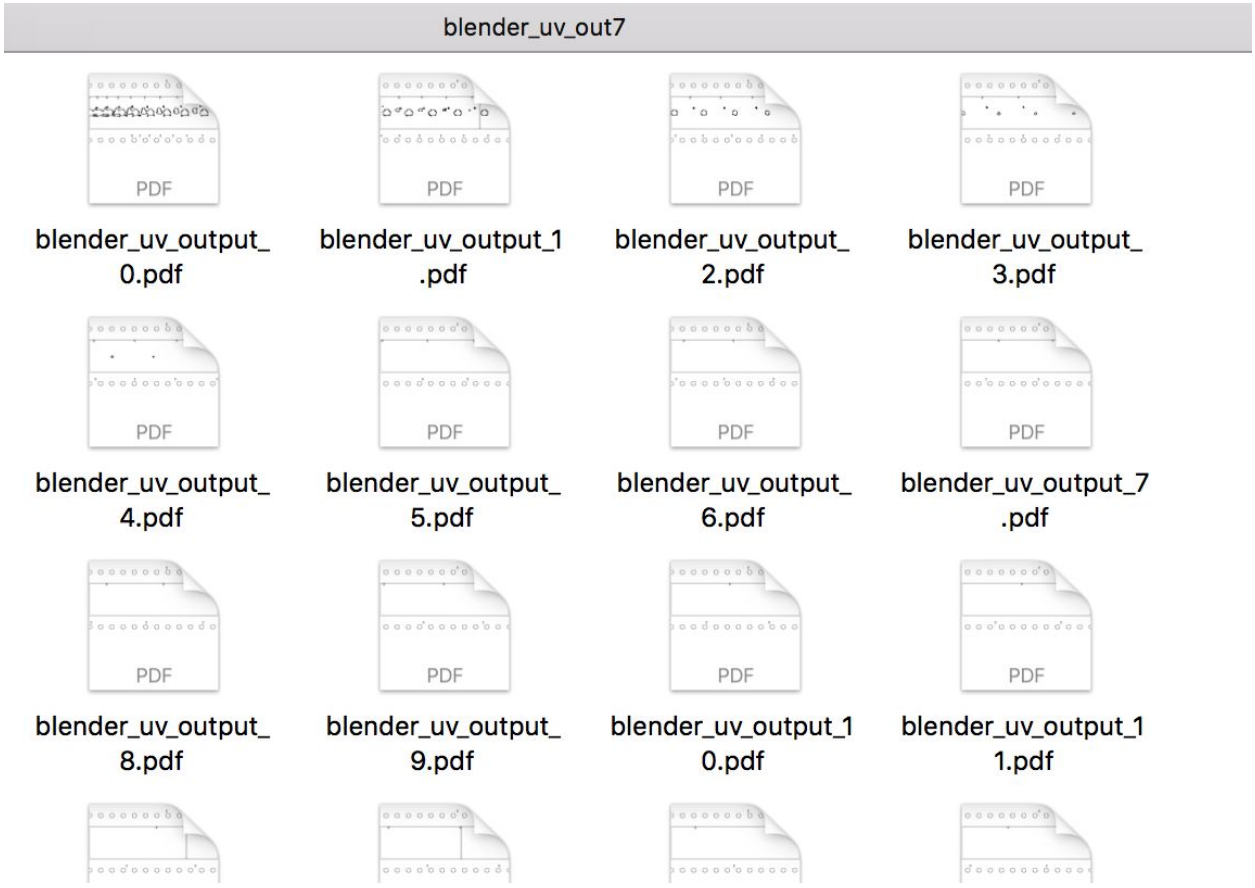


The fifth script is scale.py. The output from the former steps are the right scale but not the exactly right size. So this script modify the size of the svg files to the real size. The output is placed in “blender_uv_out5” folder.

The sixth script is transform.py. This script draws the following notches onto the svg, so that the svgs could fit the printing system.



The last script is svg2pdf.py, which converts the svg files to pdf files. The final output is placed in “blender_uv_out6” folder.



Conclusion

The final output files are the pdf files in “bender_uv_out7” folder, each with the size that the laser cutter can cut.

One assumption is that Blender app is run from the ‘/Application’ folder, which is the system default. This is because the code uses the absolute path of the shell script and need to access from the folder where the Blender app is placed. So this code only would only work on OS X.