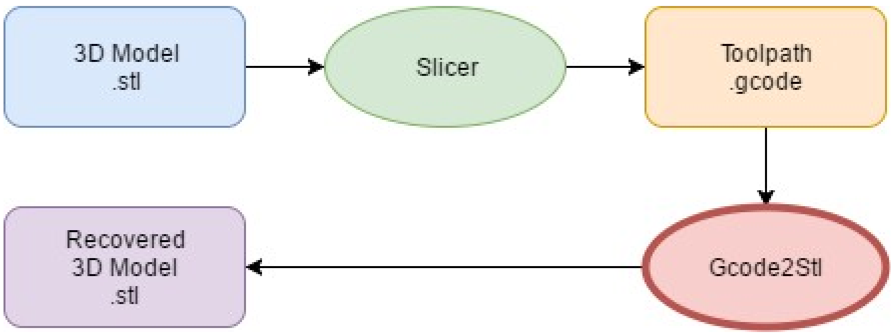
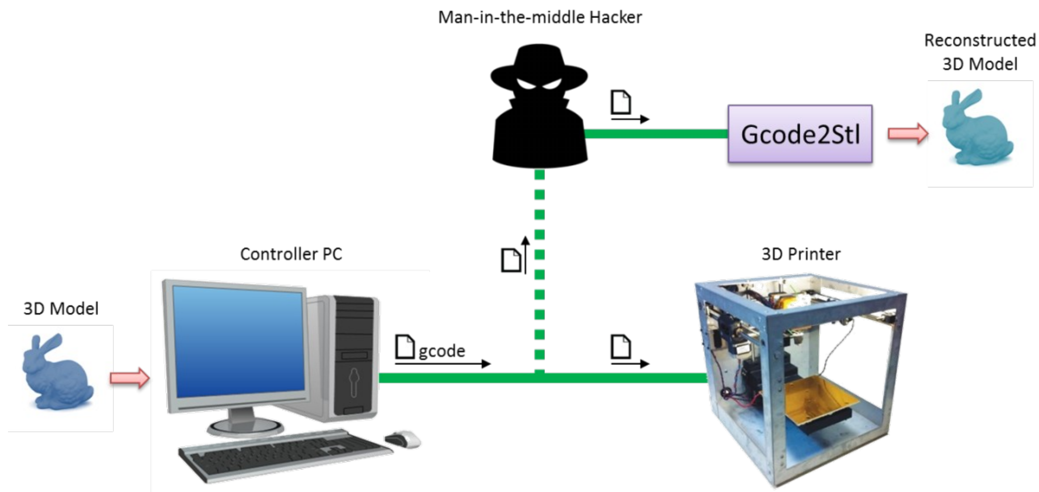
As 3D printing continues to develop, concerns arise about the security issues regarding its process, specifically, the issue of intellectual property theft of 3D models.

In the 3D Printing Process, a 3D model, usually a STL file, is translated to commands interpretable by the printer. These commands are called GCode, and the translation process, Slicing.

The goal of this project is:
Given a file containing printing commands of a model (.gcode), we would like to recover the closest possible copy of the original model (.stl).



A plausible scenario:
In a big firm, a 3D printer is connected with a cable to a controller PC. In order to protect the firm's patents (STL models), the models undergo slicing and the GCode commands are sent over the cable for printing. A hacker has gained access to the network, and wants to steal the firm's patents.

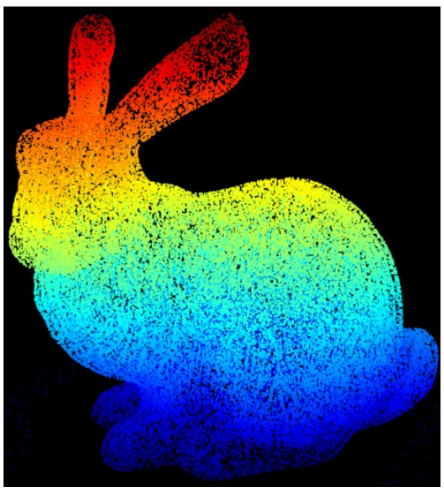


In the slicing process, a continuous 3D model (polyhedron) is transformed to printing commands using discrete sampling. As a result, there is a loss of information in the slicing process, and the reverse process is not 100% achievable.

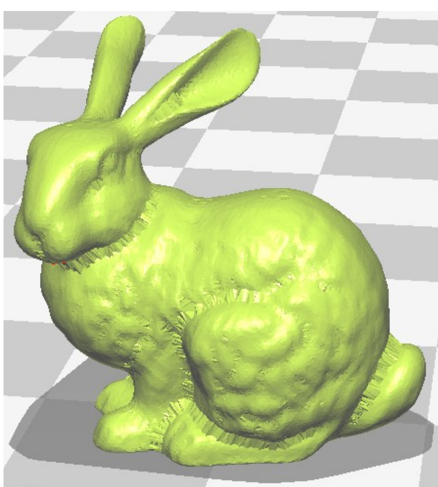
Algorithm outline:

```
;Generated with Cura_SteamEngine 2.4.0

T0
G92 E0

M109 S230
G0 F15000 X170 Y6 Z2
G280
G1 F1500 E-6.5
;LAYER_COUNT:330
;LAYER:0
M107
M204 S625
M205 X6
G1 Z4
G0 F6000 X107.338 Y98.628 Z2.27
G0 X107.848 Y98.192 Z2.27
M204 S500
M205 X5
;TYPE:SKIRT
G1 Z0.27
G1 F1500 E0
G1 F1200 X108.342 Y97.829 E0.00908
G1 X108.852 Y97.504 E0.01804
G1 X109.4 Y97.249 E0.02699
G1 X110.195 Y97.022 E0.03924
G1 X114.563 Y96.19 E0.10511
```

1: Parsing the GCode file



2: Sampling a point cloud along the GCode toolpath

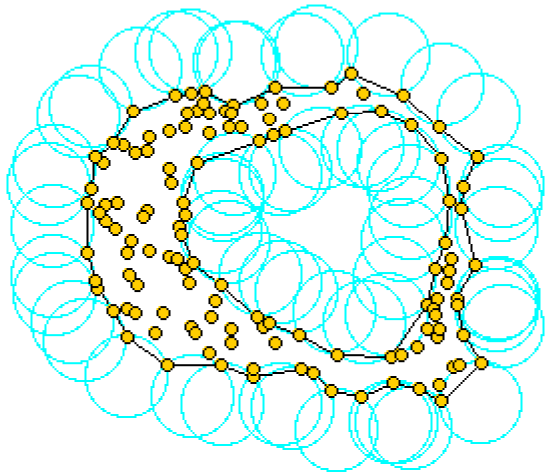


3: Reconstructing the model using α-shapes

A bit about α-shapes (from the referenced article):

Assume we are given a set $S \subset R^d$ of n points in 2D or 3D and we want to have something like “the shape formed by these points.” This is quite a vague notion and there are probably many possible interpretations, the α-shape being one of them.

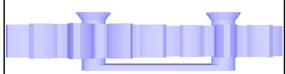
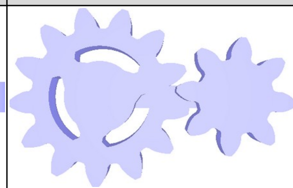
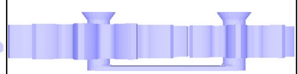
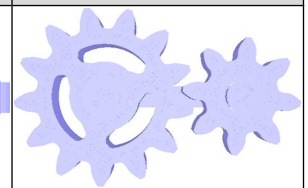
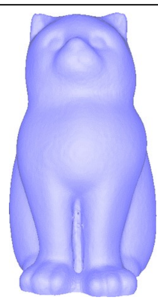
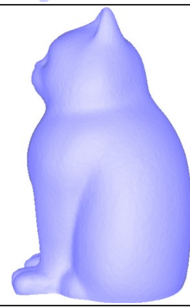
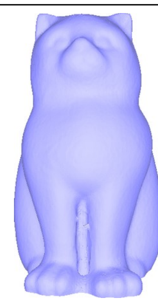
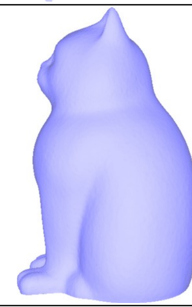
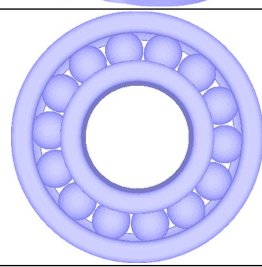
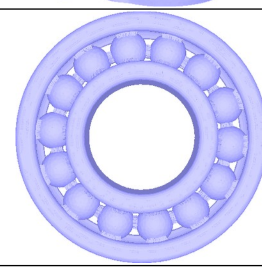
One can intuitively think of an α-shape as the following. Imagine a huge mass of ice-cream making up the R^d space and containing the points S as “hard” chocolate pieces. Using one of these sphere-formed ice-cream spoons we carve out all parts of the icecream block we can reach without bumping into chocolate pieces, thereby even carving out holes in the inside (eg. parts not reachable by simply moving the spoon from the outside). We will eventually end up with a (not necessarily convex) object bounded by caps, arcs and points. If we now straighten all “round” faces to triangles and line segments, we have an intuitive description of what is called the α-shape of S.



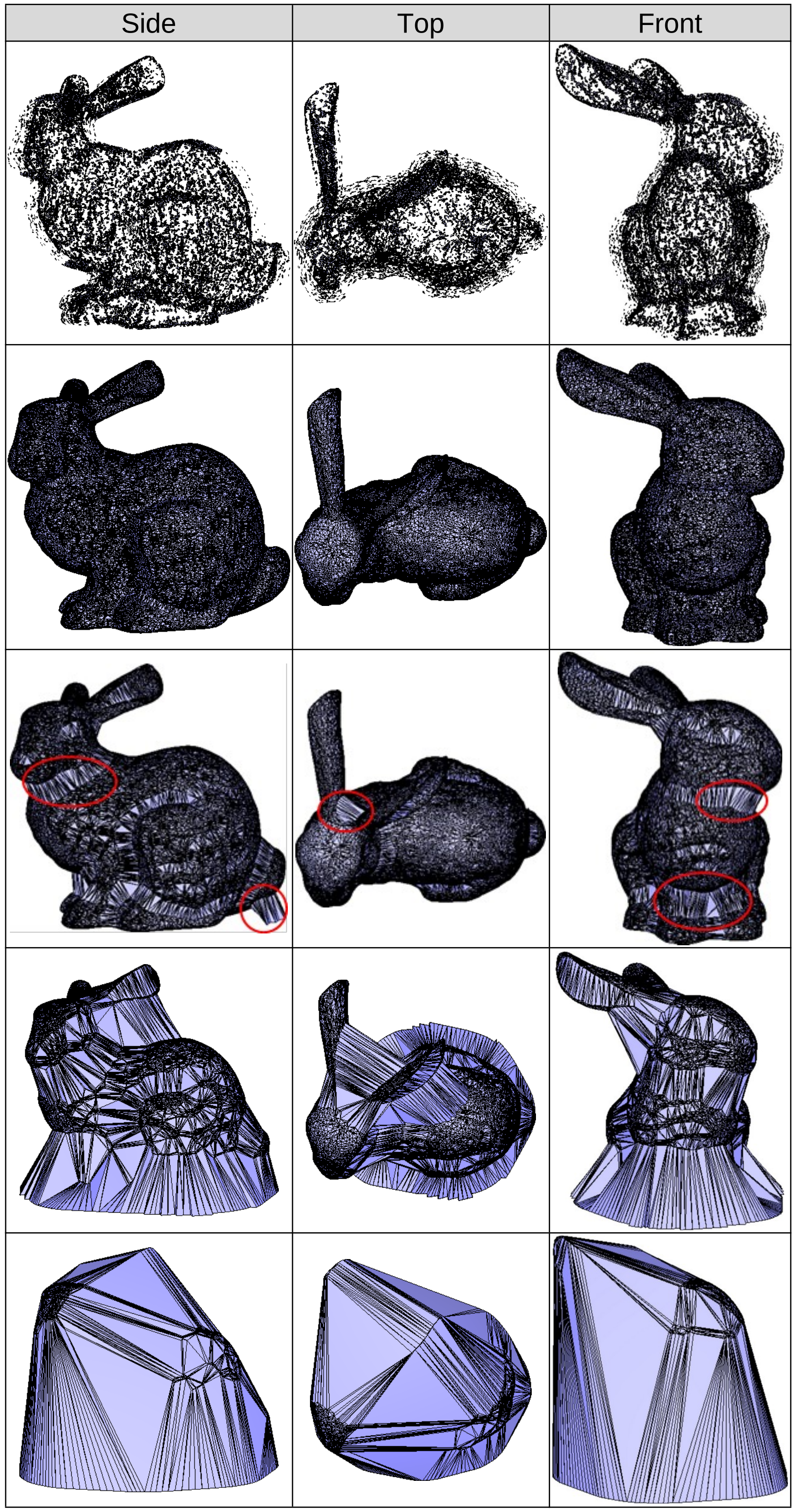
And what is α in the game? α is the radius of the carving spoon. A very small value will allow us to eat up all of the ice-cream except the chocolate points S themselves. Thus we already see that the α-shape of Sdegenerates to the point-set S for $\alpha \to 0$. On the other hand, a huge value of α will prevent us even from moving the spoon between two points since it’s way too large. So we will never spoon up ice-cream lying in the inside of the convex hull of S, and hence the α-shape for $\alpha \to \infty$ is the convex hull of S.

Results:

Models reconstructed using the α value chosen by the algorithm

Original Model		Reconstructed Model	
1	2	1	2
			
			
			

Models reconstructed with increasing α values.



References:

Introduction to Alpha Shapes. Kaspar Fischer.
https://graphics.stanford.edu/courses/cs268-11-spring/handouts/AlphaShapes/as_fisher.pdf