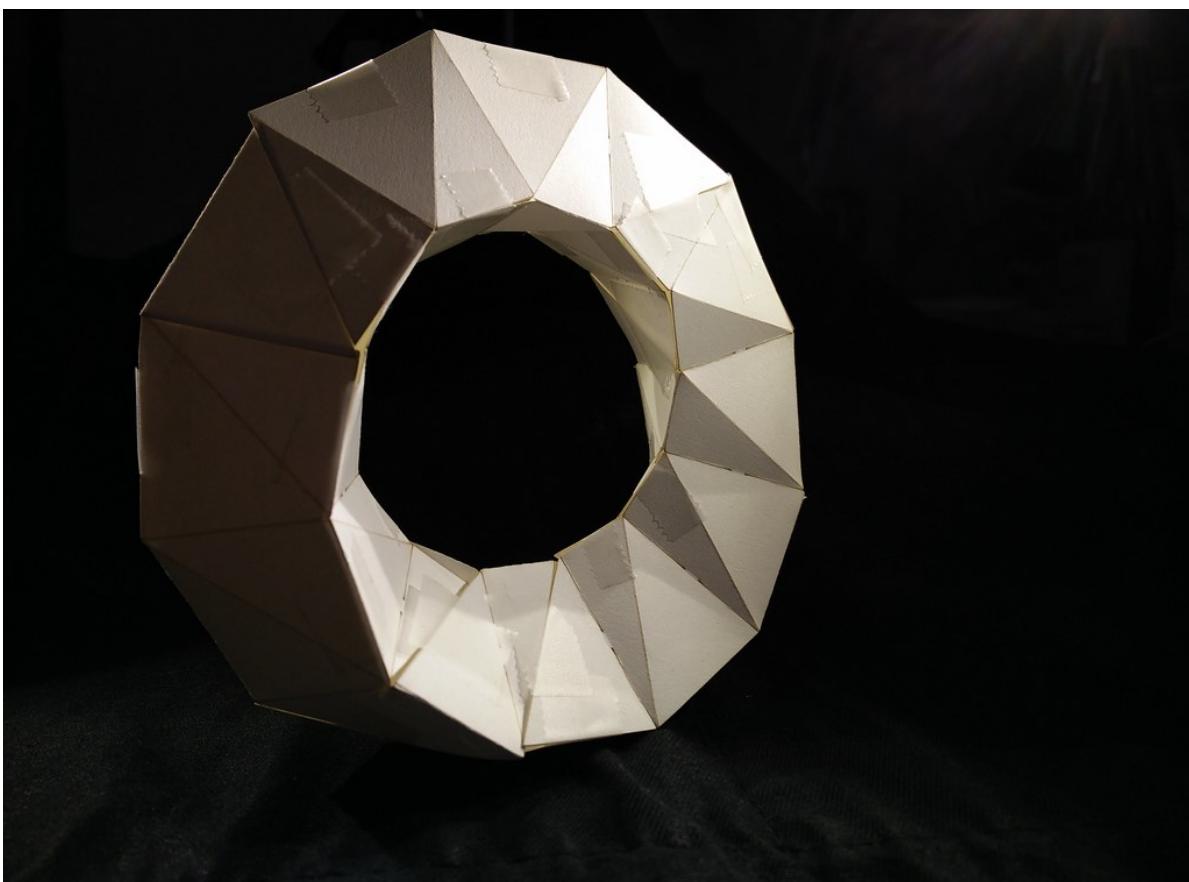
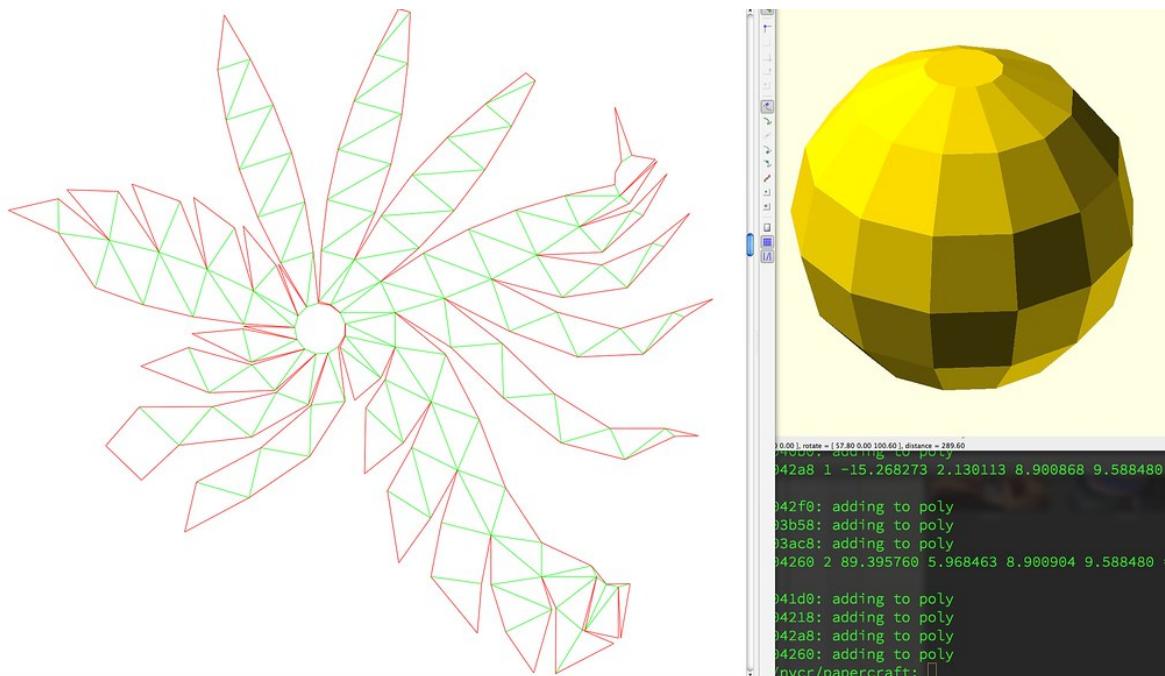


Unfolding STL



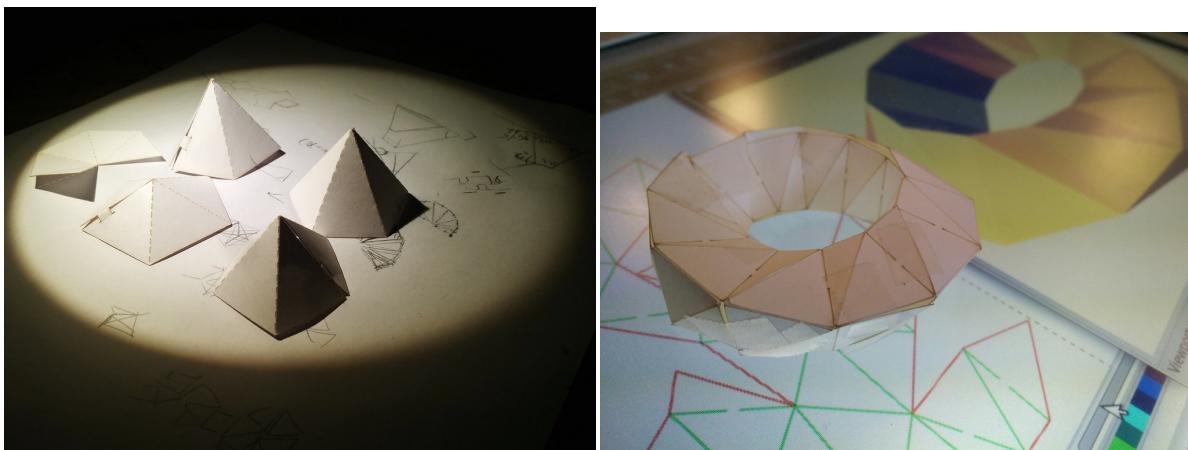
Using the power of math and trigonometry, I've been working on a way to unfold 3D STL files into sections that can be cut on the laser cutter. This is similar to [Pepakura](#), although that tool is Windows only and not Free Software. The Möbius torus ([thing:621539](#)) is my first successful model that has been converted to papercraft using my software.

Using unfold



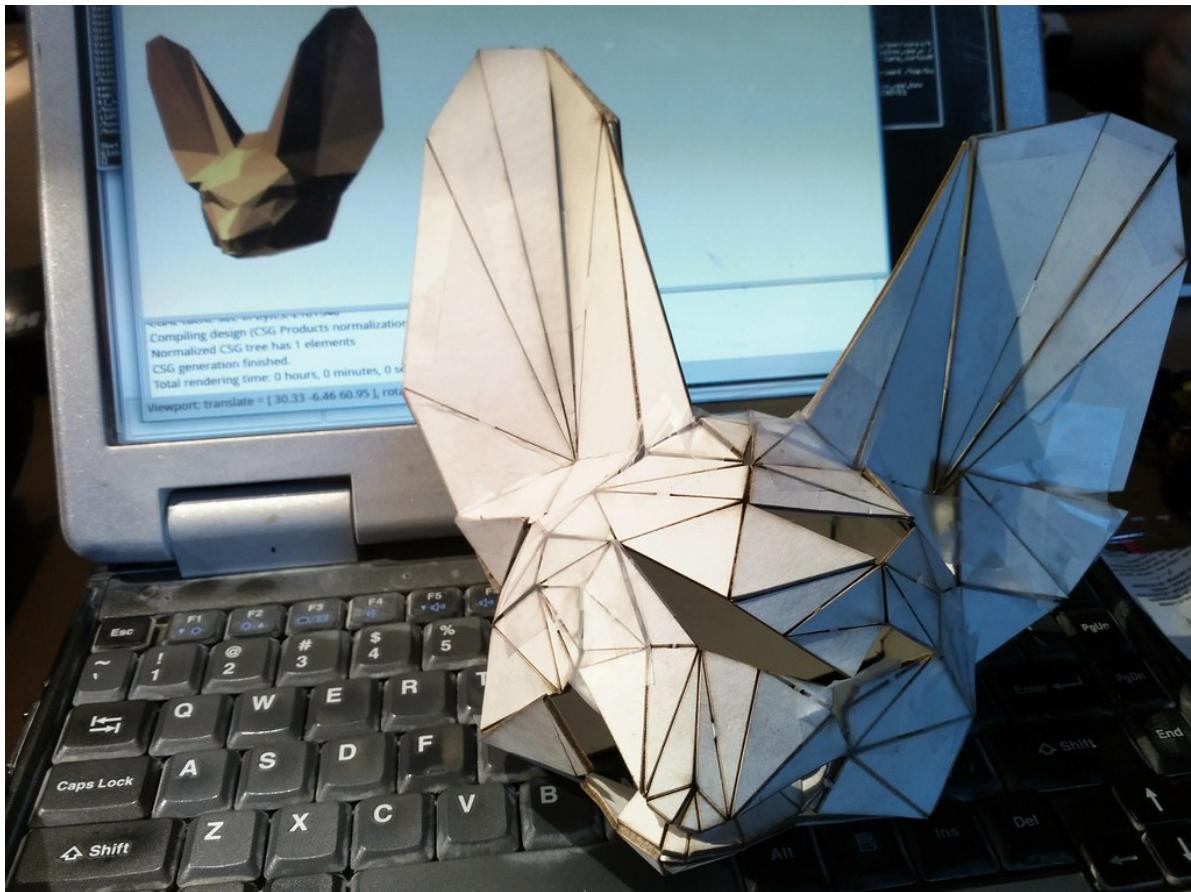
`unfold.c` doesn't have much in the way of a user-interface. It reads a **binary** STL on stdin and writes the SVG to stdout (note that OpenSCAD outputs ASCII STL files, so you can use the included `stl-convert` script to translate the ASCII to binary in a pipeline). It chooses a random triangle to start with, so multiple runs might be necessary to find the lowest number of groups. (TODO: implement some sort of search algorithm other than just a greedy collision detection)

The green lines are to be scored and are typically cut at a lower power. The dashed line indicates a "valley fold", in which the crease goes away from you, while the solid line is a "mountain fold", in which the crease comes towards you. Since the search doesn't generate an optimal folding, sometimes coplanar triangles will be split between pieces and sometimes it can be difficult to figure out which pieces attach to where. The red line is the outer cut and should be done at a higher power and done last.

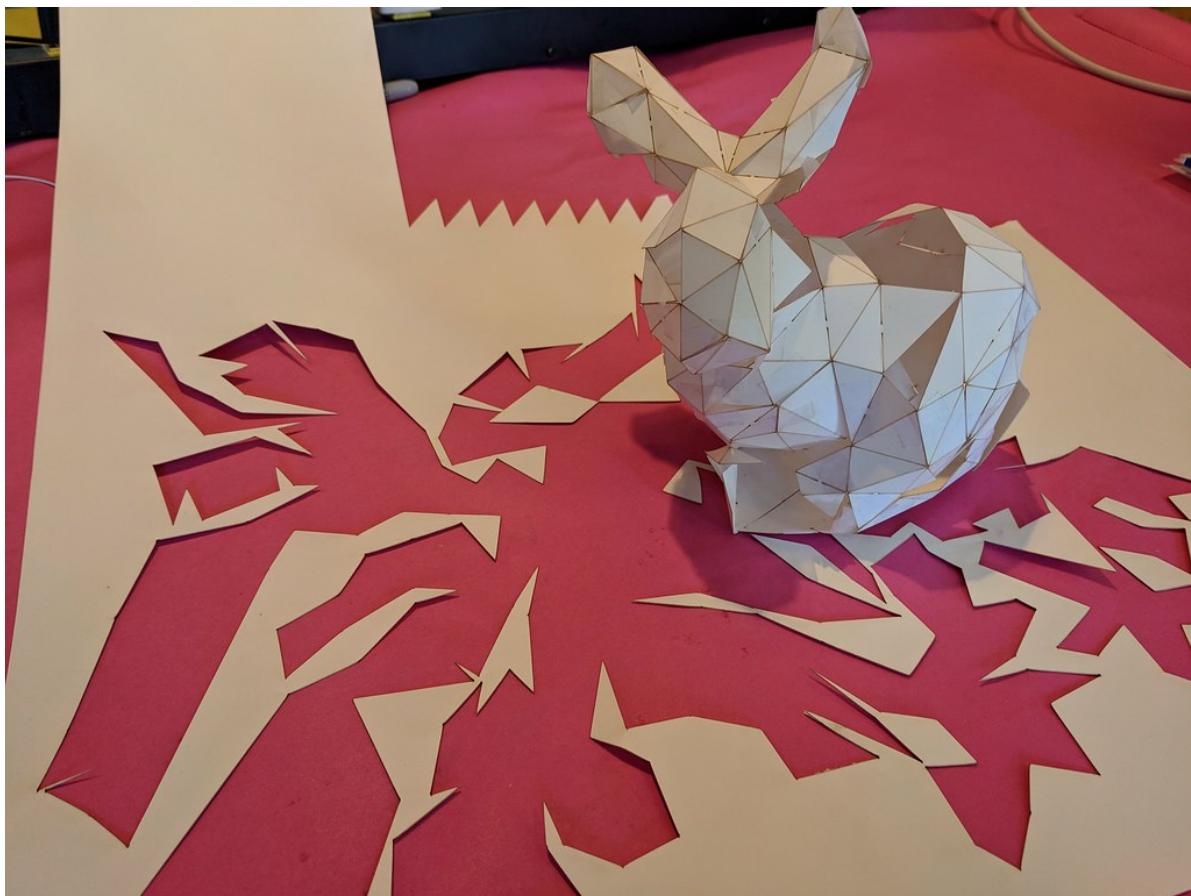


It currently works great simple geometric shapes, like these multi-faceted [cones](#) or the Möbius strip torus. The overlap detection and automatic tab generation is still very much a work in progress (there is no tab generation). Source code for the very beta version is github.com/osresearch/papercraft.

Complex STL files



Low poly shapes work ok, such as this Fennec fox. I cut it from matte board and folded it together. You can make your own, [thing:1066306](#).



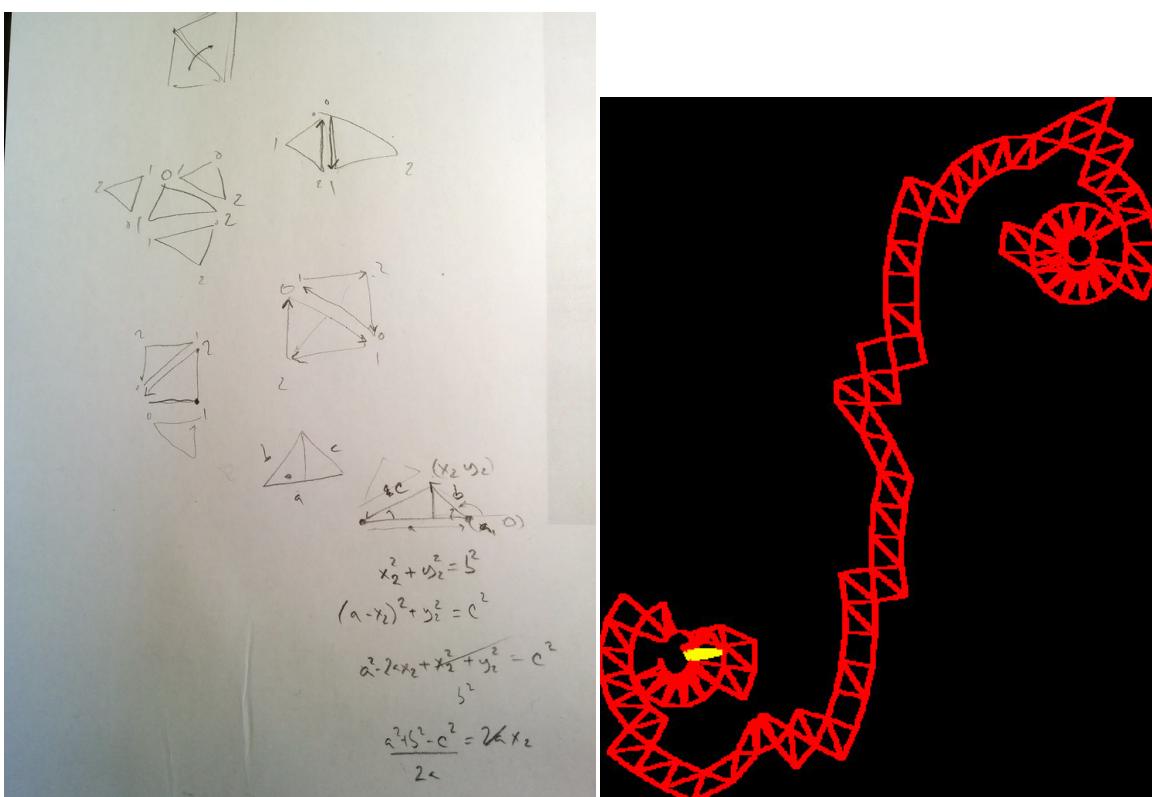
More complicated shapes, such as the [Stanford bunny \(thing:151081\)](#) are much harder to fold. I haven't successfully put this one together. One suggestion is to print a 99% version of the STL file, then wrap the paper around it, although I haven't tried that yet. Using a low poly version like [thing:2181407](#) makes it possible to unfold, resulting in [thing:3220888](#), although there are still gaps from missing triangles.



Using

large format poster board, I was able to make a Low Poly Fox. The design is [thing:3285074](#) and it has only a few missing triangles, although some of the connections to the flat bottom are *very* small edges, so it is almost easier to assemble in sections that connect to the base. I should add a way to ignore the base so that it is a hollow shape.

How it works



The math involved is mostly simple trig. The triangles in the STL file are parsed to build a graph of connections with a flag for coplanar-ness, and then a breadth-first search is done to find a way to lay the triangles flat on the sheet. Depth-first tends to produce long strings, such as the sphere immediately above, while breadth-first does a much better job of using the rectangular space. The sphere at the top of the page is breadth-first and has far less wasted exterior material.

Laser cutter | 3D Printing | OpenSCAD | 2015

Last update: November 8, 2020