# Custom 3D-Printed Rollers for Frieze Pattern Cookies

Robert Hanson
Towson University, Emeritus
Towson, MD 21252 USA
Email: rhanson@towson.edu

George Hart
http://georgehart.com
Stony Brook, NY 11790 USA
Email: george@georgehart.com

## Abstract

Inspired by ancient Near Eastern cylinder seals, we describe a method for converting images of repeating patterns, e.g., Roman friezes or Escher tessellations, into 3D-printed rollers for shaping custom-designed cookies. The rollers may be used as a math club activity to make cookies for a math event and can also be used in other creative works, e.g., imprinting ceramics and similar materials. Examples are shown of sugar cookies based on tessellation patterns of M.C. Escher, and the idea can be extended to many types of patterns and media. The 3D files and the generating software are made freely available, so anyone can download them and produce rollers for their own favorite frieze patterns.

## Introduction

Geometrically designed cookies are one of the delightful intersections between the world of mathematics and the art of cooking. Fractal cookies based on stretching and folding [1] and 3D-printed Escher-based cookie cutters [2] are two creative examples of mathematical cookie ideas that can be found online. Home-baked cookies and mathematical patterns are widely appreciated, but tastes differ as to what is the best kind of cookie and what are the most interesting geometric patterns, so we see value in a technique that can be adapted to many tastes.

This paper provides a customizable method of producing cookies that are imprinted with an individual's favorite frieze patterns and tessellations, as in Figure 1. Custom textured rollers are used to transfer a pattern to the cookie dough surface before baking. The rollers are produced individually on a 3D printer, which may be accessed at a school, an online service bureau, a “fab lab,” or a hobbyist's home. To drive the 3D printer, a geometry description file (called an “STL file”) is needed; it specifies the exact shape of the object to be built. The software described below outputs the necessary file for any design the user specifies. The design is specified by inputting an image file to the software, which converts darkness in the image to surface depth in the roller. A wide variety of images can be found online, and image processing software allows users to easily modify these images and create original ones, so this process gives users complete control of their design without needing expertise in 3D design software. The software is provided free online [3] and in the Appendix for anyone to use who has access to the Mathematica programming environment [4]. The transformations are very straightforward, so the idea could easily be adapted into other programming languages.

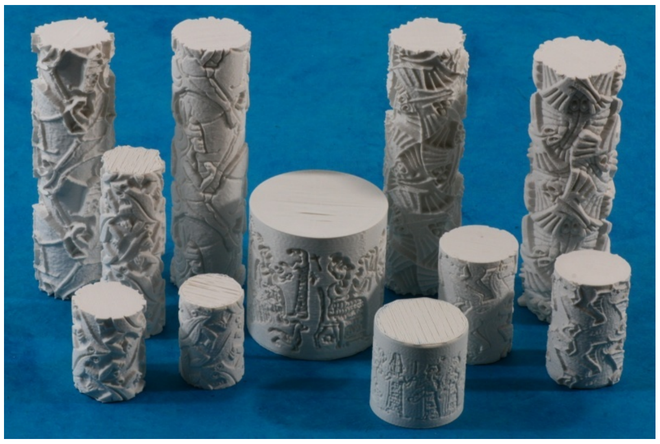
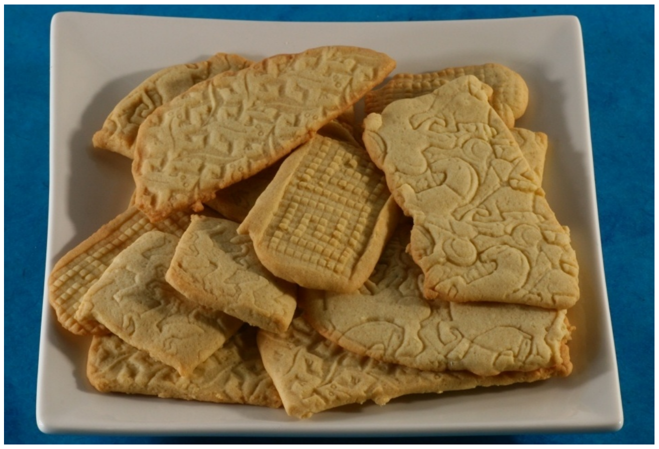

**Figure 1:** *3D-printed rollers (4 to 12 cm high) and sugar cookies with imprinted patterns.*

## Examples

These rollers are 4 to 12 cm high (when stood on end as shown) and are built from ABS plastic on a Makerbot "Replicator" 3D printer [5]. The STL files are freely available [3] so anyone can make copies of these rollers.

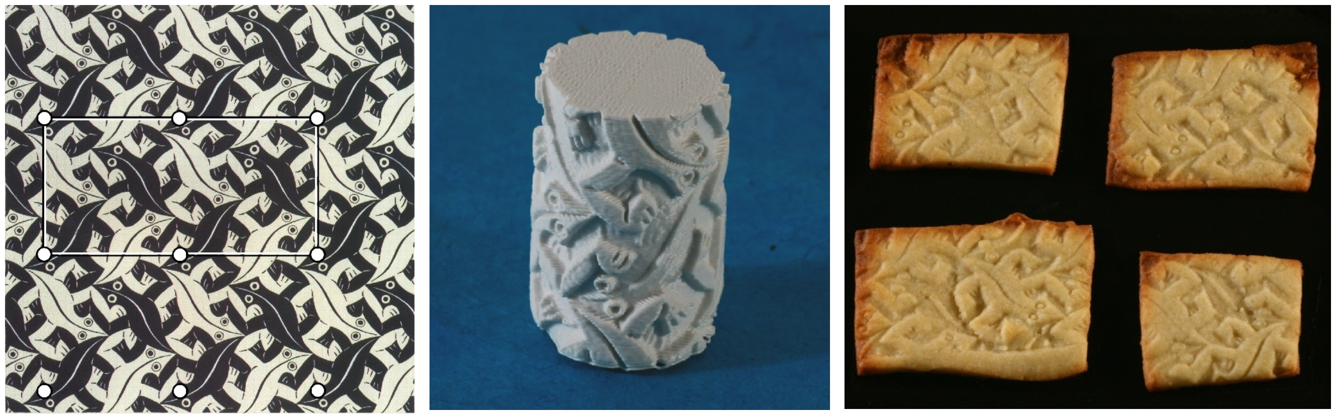

**Figure 2:** *Escher image, "Lizard," 3D-printed roller, and sugar cookies.*

The Escher image in Figure 2 consists of black and white reptiles [6]. A set of equivalent points are marked here with circles. The chosen point is the upper eye of a downward facing black reptile, but any point would do equally well to bound a fundamental region. A square bounded by four of these points could be used as a rubber stamp to produce the entire image using only translation. Because a square rolls up into a thin cylindrical shell that can be difficult to manipulate, the marked rectangle consisting of two squares was input to the software to produce the roller in Figure 2. It has two copies of the image, and twice the diameter of a roller based on a single square. The black and white regions of the image are translated into two levels of depth. A sugar cookie dough recipe [7] was rolled out and imprinted with the roller to make the cookies shown.

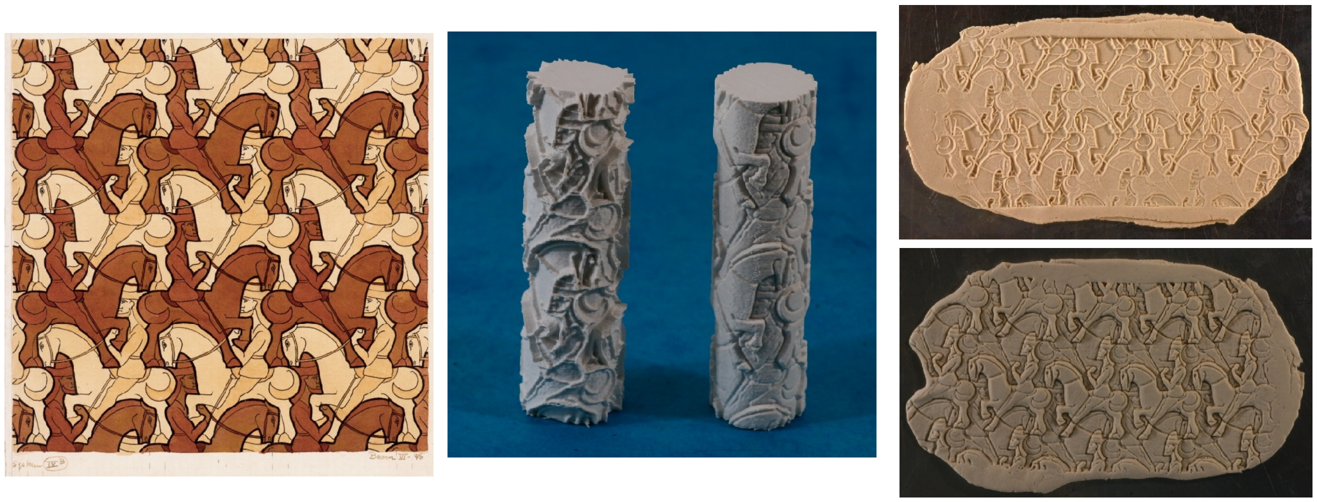

**Figure 3:** *Escher "Horsemen," positive and negative 3D printed rollers, and their Play-Doh impressions.*

In a similar way, a rectangular region from M.C. Escher's *Horsemen*, shown in Figure 3 [6], was fed to the software to produce the two rollers shown. The four levels of darkness in the image are translated to four different heights in the rollers. They differ according to whether the darkest parts of the image correspond to "mountains" or "valleys" on the roller. The user can adjust this by negating a parameter, *alpha*, described below. For this design, it is apparent that the second impression is easier to recognize, where the black lines correspond to mountains on the roller, which make valleys on the cookie surface.

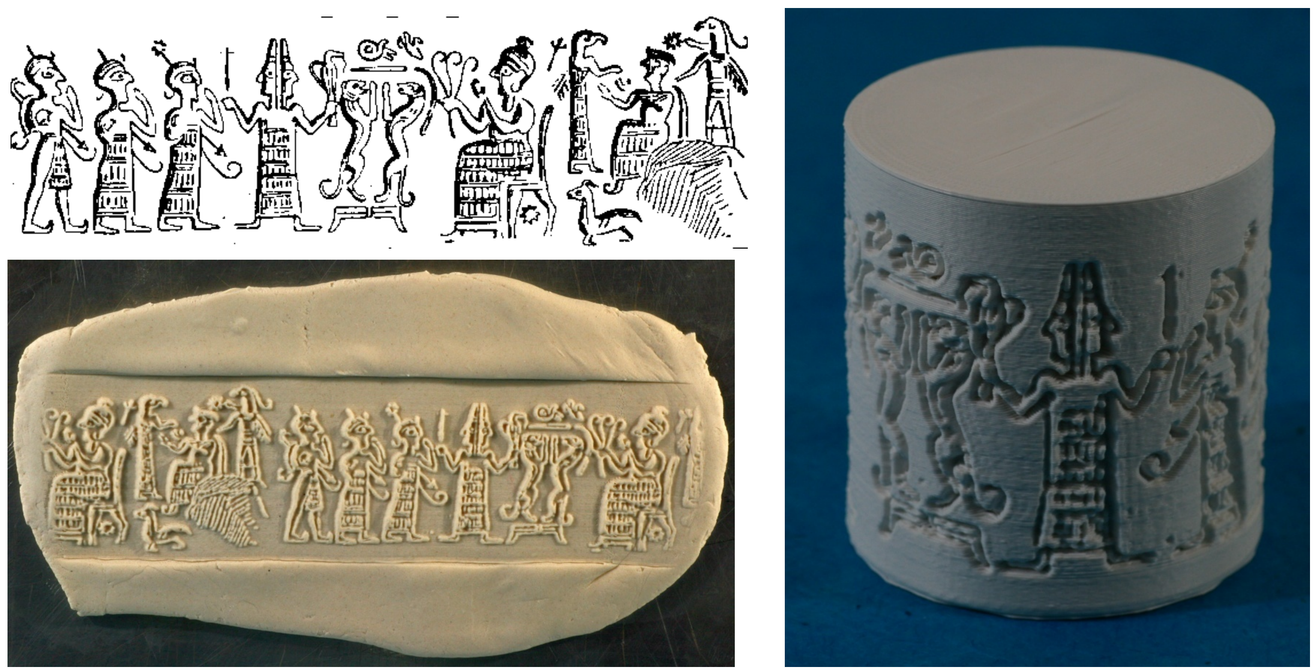

**Figure 4:** *Hittite cylinder seal image with Janus, 3D printed roller, and Play-Doh impression.*

There is a long history of carved stone rollers being used to make clay seals, e.g., to authenticate royal documents [8]. Numerous examples of intricately carved Sumerian, Hittite, or Mesopotamian cylinder seals going back to 3500BC can be found in museum collections. An exhibition of ancient Near Eastern cylinder seals at the Morgan Library in NY inspired us to begin this project. Figure 4 shows a drawing of a Hittite seal [9], a 3D-printed roller we derived from it, and a Play-Doh impression from the roller. In this case, we started from a modern line drawing of the seal, which indicates the contours of the figures, so our cylinder recreates these lines. However, the original 15th century BCE seal in the Louvre Museum [10] on which the drawing is based is actually a relief in the style of the following example.

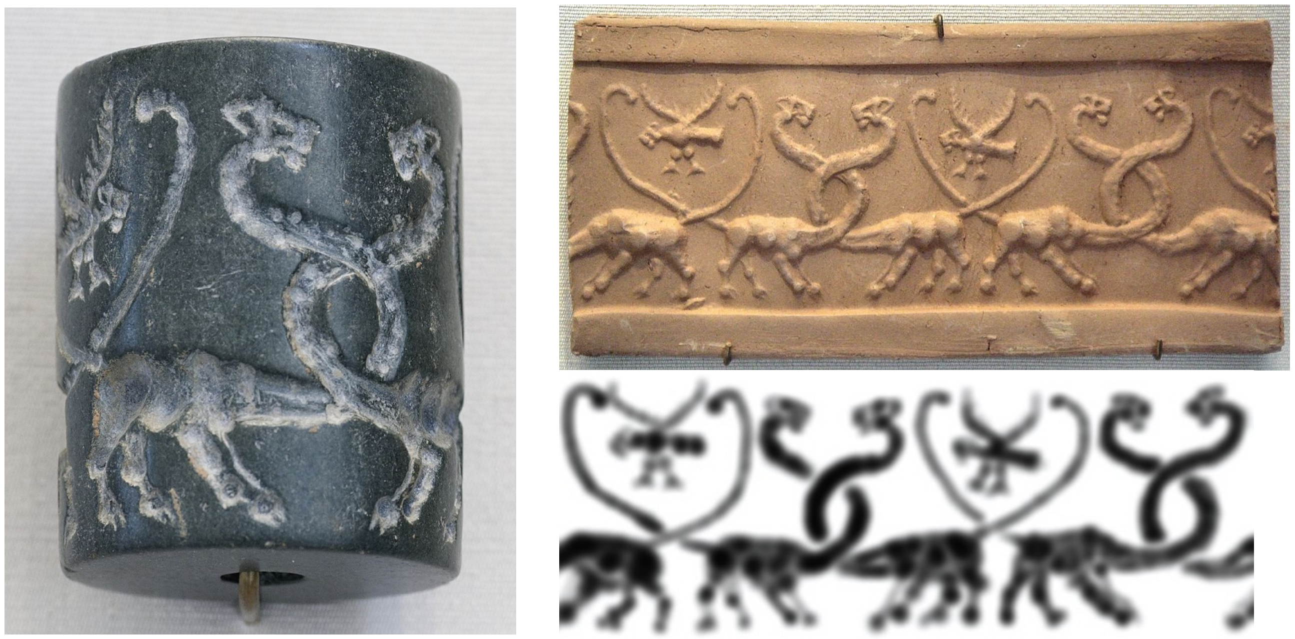

**Figure 5:** *Mesopotamian cylinder seal, modern impression, and "depth map" reconstruction of one cycle.*

Figure 5 shows an interesting Mesopotamian seal from 3000BCE, also in the Louvre Museum, and a modern impression taken from it [11]. Some people promote this example as evidence that dinosaurs and humans once co-existed, but we (and no doubt the Louvre) do not endorse that interpretation. Unlike the line drawing of Figure 4, this image makes clear the nature of the typical reliefs on ancient cylinder seals. The depth of the carving is a sculptural representation of the volume of the objects shown. To recreate this style, called "intaglio" or "counter-relief," we need to feed the software an image in which darkness corresponds to depth. In modern computer-oriented jargon, this is termed a "depth map". An approximate reconstruction of its depth map, shown at bottom right in Figure 5, was made by tracing the impression image using an air-brush tool in a drawing program. Feeding it to the software allows us to produce the roller in Figure 6 and recreate the relief style of the impression. Close examination shows this is not a very exact reconstruction, but it illustrates a possible technique if one wishes to use this software to produce this style of roller.

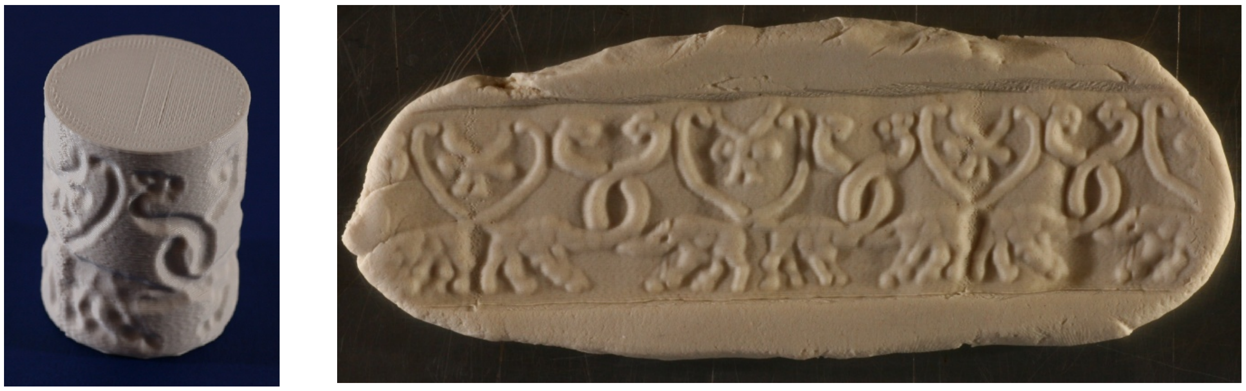

**Figure 6:** *3D-printed cylinder and Play-Doh impression, roughly reconstructing Figure 5.*

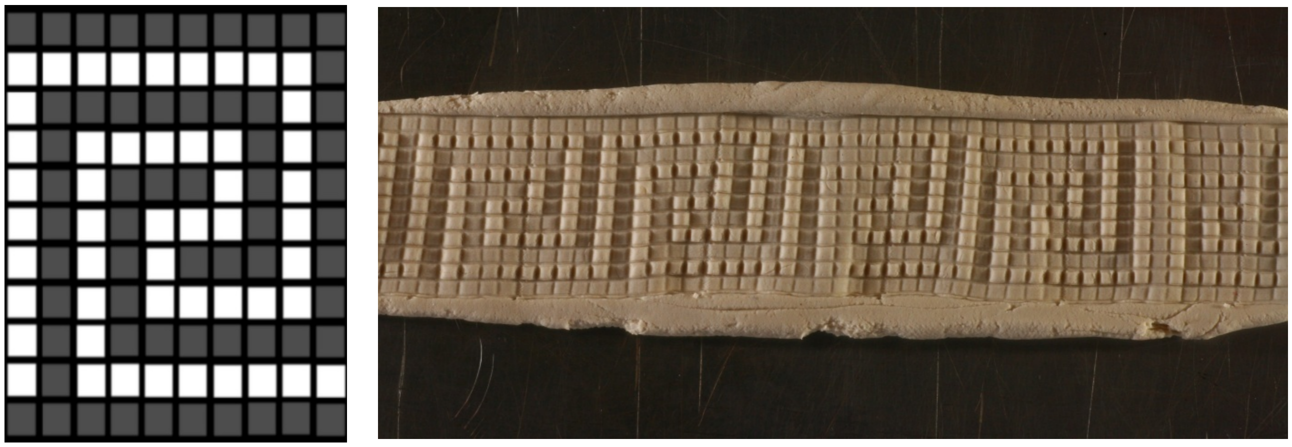

**Figure 7:** *Greek Key mosaic unit drawing and resulting frieze pattern imprinted in Play-Doh.*

Rollers for arbitrary frieze patterns can be produced by creating one translational unit of the pattern in a drawing program and feeding the image to the software. Figure 7 shows an example in which we created a mosaic-like image by stamping white and grey squares on a black background to create one unit of a classic "Greek Key" frieze pattern. The resulting roller has two heights of square tiles rising above the lower background of "grout" between the tiles. Some cookies made from this roller can be seen in Figure 1.

## Geometry

The software implements mathematical transformations of linear scaling (darkness to distance) and changes of coordinate systems. An image file provides a table of data which encodes darkness as a function of *x* and *y* position in the image. The program transforms *x* and *y* to become angle and height in the cylindrical coordinate system, with the full range of *x* scaled to one complete revolution, so the right and left sides of the image join.

The nominal radius of the cylinder is chosen so that the ratio of the cylinder's height to its circumference equals the ratio of the image height to its width. Then this nominal radius is modified at each point by adding an increment proportional to the darkness of the image at the corresponding pixel.

An arbitrary parameter, called *alpha* in the program, can be specified by the user to scale this depth increment. Setting a larger value for *alpha* causes the white regions of the image to be carved deeper into the cylinder, leaving the black to protrude from the cylinder and make deeper indents in the cookies. Setting *alpha* to zero would give a smooth cylinder with no texture, so is not useful. Setting *alpha* negative reverses the direction of the increment, so white regions of the image protrude from the cylinder, which gives a different visual effect. Figure 3 shows the effect of negating *alpha*. Most users with drawings containing dark lines will find a positive *alpha* works best, but some experimentation may be worthwhile.

Finally, the cylindrical coordinates are transformed into 3D Cartesian coordinates which are exported to the STL file. The handedness is such that the image appears reversed on the cylinder, so it is not reversed in the cookie. Circular caps are added at the top and bottom of the cylinder so the whole is a "water-tight" manifold without boundary, which is necessary for the slicing software in 3D printers to operate correctly.

## Process

To use the software for your own custom roller, follow these steps:

1. Find or create an image file in any standard image format. Degrees of darkness will correspond to desired depth, so it is best if it has good contrast. If starting with a color image, converting it to gray-scale in an image editing program will show how the software interprets it. The X direction of the image will wrap around the circumference of the roller and should be about 400-500 pixels wide for reasonable resolution without an overly large file. Rescale the image if necessary. It may be useful to blur or smooth the image if it is grainy, to avoid a very jagged roller surface. If working from an image of a frieze pattern or tessellation, crop it to a rectangle that repeats by translation, as indicated in Figure 2. Save the file in any standard image format.

2. Download the software [3] and open it within Mathematica. Execute the first block of code to import your image. Edit the lines which specify the desired cylinder height and the *alpha* parameter if you wish to change them. The height is specified in arbitrary units that are set in the 3D printer's software, commonly inches in the US and mm elsewhere. *Alpha* is the fraction of the height that black is scaled to, e.g., if *alpha*=0.1, the deepest indents of a 2-inch high roller will be 0.2 inches. A negative *alpha* results in dark portions of the image indented in the roller and light portions sticking out. Execute the second block of code to generate and display the cylinder model. You will probably need to adjust *alpha* and regenerate it until the texture scale looks appropriate. When the scale of the mountains and valleys looks OK for your application, execute the final line of code to generate the STL file. (Set the output file name as you wish, but include the ".stl" file extension.)

3. Send the STL file that is generated to your 3D printer or an online 3D printing service bureau to produce the cylinder. You will need to specify what units (inches or mm) to use.

4. Make cookie dough according to your favorite recipe and roll the roller over it. Cookies which rise significantly during baking will lose some of the imprinted detail. We find a simple sugar cookie works well. If desired, decorate them further by filling the depressions or coloring the raised areas. Bake and enjoy!

## Applications and Future Work

Math cookies are good any time, but they may be especially valuable in providing a fun activity for a classroom or math club. Many aspects of symmetry and transformations may arise, giving the teacher an opportunity to informally introduce the underlying mathematical ideas. The seven classes of frieze patterns, as categorized by their symmetries, are a natural topic to discuss. It should be pointed out that only the translation operation is naturally captured by the process of rolling a cylinder. Any additional symmetries of a frieze, e.g., reflections, rotations, or glide reflections, can be incorporated by explicitly repeating the pattern on the cylinder surface.

Custom rollers are also useful to ceramicists for applying repeating patterns in their work. There is a sensual pleasure to be found in working with clay, and we find it is very satisfying to roll these patterns into Play-Doh and physically recreate both ancient designs and Escher's designs. Our current experiments with rolling soft clay, impressing patterns, and baking are very promising. A software option which we have implemented but not illustrated here creates a cylindrical hole along the axis of the roller. This mimics certain ancient seals that were drilled out so they could be worn as an amulet necklace. But we implemented it also to allow ceramicists to place a dowel through the roller, to use as a handle for finer control of the clay.

Future work could look at software techniques to automate parts of the process, for example, finding the optimal value of *alpha* for a given image or cropping an optimal unit rectangle from a tessellation image. A friendlier user interface might make it easy to double or treble an image, to avoid making a roller which is too thin to roll easily, as we did manually when using two squares in Figure 2. More advanced culinary research might model the dynamics of how cookie dough expands when baked, and adjust the roller height field to create imprints for optimal readability after cooking.

## Conclusion

Carl Sagan once said that “if you want to make an apple pie from scratch, you must first invent the universe.” We see a corollary that if you wish to make cookies from scratch you might first need to write software to generate the cookie roller's geometry. This paper provides that step, so users can go on to use it in making their own mathematically patterned cookies. And more importantly, we provide a general tool which others may use in many creative mathematically themed projects. We expect it will prove especially valuable to people who are comfortable with image processing software but are unfamiliar with 3D design software. We look forward to seeing what others create using their own 3D-printed roller designs.

## References

[1] Evil Mad Scientist Laboratories, http://www.evilmadscientist.com/2008/sierpinski-cookies/
[2] Thingiverse, http://www.thingiverse.com/thing:3248
[3] G. Hart, http://georgehart.com/rollers
[4] Wolfram Research, http://wolfram.com/
[5] Makerbot, http://makerbot.com
[6] Doris Schattschneider, *M. C. Escher: Visions of Symmetry*, Abrams, 2004
[7] G. Hart, trilobite recipe, http://www.georgehart.com/trilobites/trilobite.html
[8] Wikipedia, http://en.wikipedia.org/wiki/Cylinder_seal
[9] Bill Schell, http://campus.murraystate.edu/academic/faculty/bill.schell/civweb101/Hittites.htm
[10] Louvre cylinder photo, http://www.lessing-photo.com/dispimg.asp?i=08021530
[11] Wikimedia, http://commons.wikimedia.org/wiki/File:Uruk3000BCE.jpg and
http://commons.wikimedia.org/wiki/File:Cylinder_seal_lions_Louvre_MNB1167_n2.jpg

## Appendix

For completeness, the executable statements of the Mathematica code are given here. To save space, comments and other material are omitted, but the complete executable file in Mathematica's format can be downloaded [3].

```
img = ColorConvert[Import[SystemDialogInput["FileOpen"]],"Grayscale"]

height = 2.0;
alpha = 0.05;
{n,m} = ImageDimensions[img];
rMid = n * height/(2 Pi m);
r = Map[rMid - alpha height(#-0.5)&,ImageData[img],{2}];
xyz = Table[{r[[j,i]]Sin[2 Pi i/n],r[[j,i]]Cos[2 Pi i/n],(1-j/m) height},{i,1,n},{j,1,m}];
surface = Table[{xyz[[i,j]],xyz[[Mod[i+1,n,1],j]],
                 xyz[[Mod[i+1,n,1],j+1]],xyz[[i,j+1]]},{i,1,n},{j,1,m-1}];
bottom = Table[{{0,0,0},xyz[[i,m]],xyz[[Mod[i+1,n,1],m]]},{i,1,n}];
top = Table[{{0,0,(1-1/m)height},xyz[[Mod[i+1,n,1],1]],xyz[[i,1]]},{i,1,n}];
g = Graphics3D[{EdgeForm[],Map[Polygon,Join[surface,top,bottom]]}]

Export[SystemDialogInput["FileSave","roller.stl"],g]
```