A Construction of Mathematical Art

Paul Gafni

March 2017

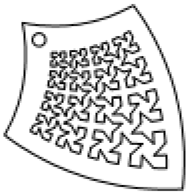

Figure 1: A Fractal Pendant

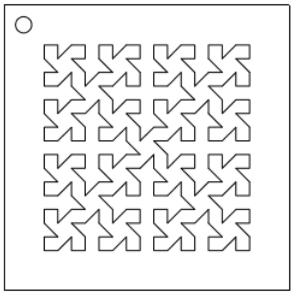

Figure 2: A Simpler Fractal Pendant

1 Introduction

In this paper, we discuss the construction of Figure 1, using Rhino 5 and Grasshopper, in order to empower you to create your own art. Figure 1 is a distortion of Figure 2. In Section 2, we outline the construction of Figure 2, using Rhino 5. Then, in Section 3, we present and describe the Grasshopper script that turns Figure 2 into Figure 1. Finally, in Section 4, we give some commentary on the mathematical significance of the Fractal in Figure 2 and on the transformation given in Section 3.

2 Using Rhino 5 to Construct Figure 2

We take advantage of the self-similarity of Figure 2 in order to expedite the construction, relying on repeated use of the POLARARRAY command. We start by using the PLINE command to create the shape shown in Figure 3.
We use the POLARARRAY command, centered at $(1.5, 1.5)$ to create Figure 4.
Using the LINE command, we connect the four pieces with diagonal lines in order to make a single polygon. It is useful to have OSNAP (object snap) set to enable snapping to endpoints. Then, we delete a single edge in the top right corner. Since the edge in the top right corner is part of a PLINE, we first EXPLODE that

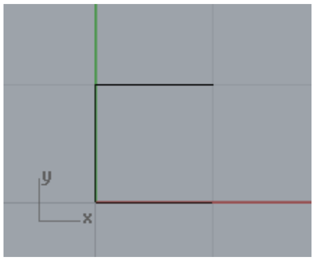

Figure 3: PLINE[(1,1)–(0,1)–(0,0)–(1,0)]

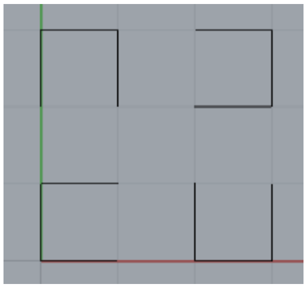

Figure 4: POLARARRAY[Center= $(\frac{1}{2}, \frac{1}{2})$, n=4, Angle Fill= 360°]

PLINE into 3 individual segments. Then we use DELETE in order to create Figure 5. As a matter of taste, we may choose to GROUP all of these objects into a single object before proceeding.

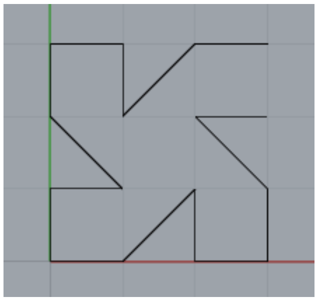

Figure 5: Using LINE, EXPLODE, and DELETE

At this point, we are ready to rinse and repeat our procedure. We use the POLARARRAY command to create Figure 6. Then, we use LINE, EXPLODE, and DELETE to create Figure 7.
One more round of POLARARRAY, LINE, and EXPLODE yields the interior of Figure 2. The border was created using the PLINE command, and the circular hole was created using the CIRCLE command. The sizing and placement of the circle included some tweaking to taste. I used a temporary line from the top left corner of the fractal to the top left corner of the border in order to ensure the circle was placed symmetrically. In the next section, we turn our attention toward transforming Figure 2 into Figure 1.

3 Using Grashopper to Transform Figure 2 into Figure 1

We begin by presenting the Grasshopper script in full, shown in Figure 8, and then we walk through each component of the script. We use the word “component” to refer to each of the individual boxes shown in the Grasshopper script. This Grasshopper script starts by using the DIVIDECURVE command to mark lots and lots of points along the entirety of Figure 2, using a SLIDER to specify the number of points. The DIVIDECURVE command outputs points in Cartesian coordinates, so we use the TO POLAR command

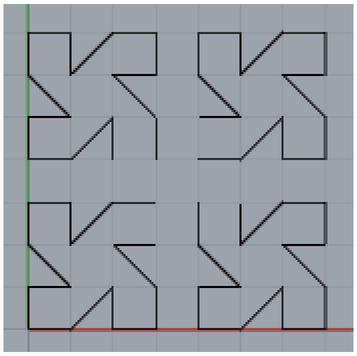

Figure 6: Another use of POLARARRAY, this time centered at (3.5,3.5)

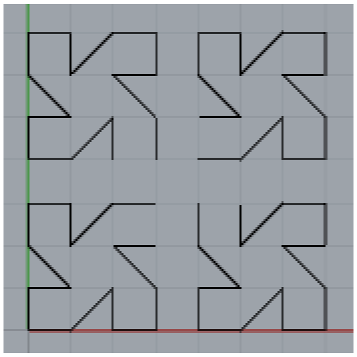

Figure 7: Another round of LINE, EXPLODE, and DELETE

in order to convert these Cartesian coordinates into POLAR coordinates. We then apply the function $f(r, \theta) = \left(\frac{1}{r}, \theta\right)$ using the DIVIDE command in Grasshopper: this is the actual distortion effect.
To put these points back together into a new fractal pattern, we convert the points back to Cartesian coordinates using the POINT POLAR command, and then use the PLINE command to construct a new PLINE. Amazingly, Grasshopper manages to keep track of the relationships between the points, and puts together the new PLINE exactly as we'd like. Scaling the slider will determine the precision of the resulting image: more points will create a smoother result.
We turn the "Preview" off on each of the components except the last one, in order to keep our screen manageable. Finally, we right click the final PLINE component and select "BAKE" in order to actually create the final object in Rhino. The result of the script depends substantially on where Figure 2 lies in the plane. We placed Figure 2 in Quadrant IV, overlapping the unit circle, in order to create Figure 1. Creating the border in Figure 1 involved tweaking and experimentation which are not described in this paper.

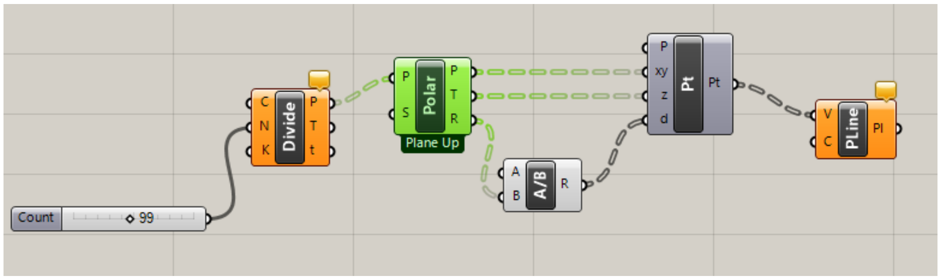

Figure 8: A Unit Circle Inversion in Grasshopper

4 Mathematical Commentary

4.1 The Mathematical Context of Figure 2

Figure 2 shows a 256-sided shape on a 16x16 array of points. It is not hard to see that it is impossible to make more than 256 sides, assuming each vertex must land on one of the points in the array. In particular, a polygon with n sides has exactly n vertices; it's clear that no more than 256 vertices are possible, since there are only 256 dots. I first discovered this fractal in analyzing exactly that question: "Given an nxn grid of dots, what is the maximal number of sides on a polygon, with the constraint that each vertex is given by one of the dots?" This fractal makes clear that it is possible to create a polygon with n^2 sides on an nxn grid, when n is a power of 2. Clearly, a 9 sided shape on 9 dots is impossible. I suspect that on a 5x5 grid, 24 sides is the best possible. If you can prove or disprove this, please contact me at GafniP@uw.edu.

This fractal is closely related to both the Hilbert Curve and the Sierpinski Curve (shown in Figures 9 and 10), but the author has been unable to find any reference to this particular fractal. It seems clear that this fractal must already have a name, but readers are invited to call it the Gafni Curve until another name is identified. If you know another name for this fractal, or if you've seen this fractal referenced anywhere, please contact me.

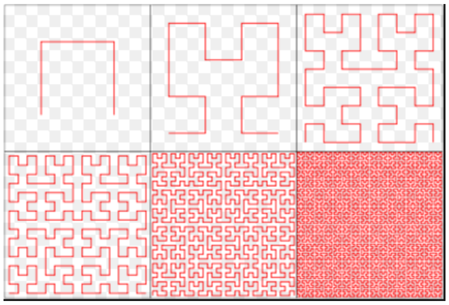

Figure 9: A Few Iterations of the Hilbert Curve

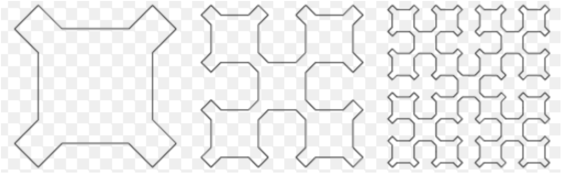

Figure 10: A Few Iterations of the Sierpinski Curve

4.2 The Mathematical Context of Transforming Figure 2 into Figure 1

The reader may have noticed that this Grasshopper script implements our familiar unit circle inversion. In polar coordinates, we can frame this as $f(r,\theta) = \left(\frac{1}{r}, \theta\right)$. As a function over the complex plane, we can frame this as $g(z) = \frac{z}{|z|^2}$.