Techniques for Artistically Rendering Space-Filling Curves

L. Kerry Mitchell
Mathematics Department
University of Advancing Computer Technology
2625 West Baseline Road
Tempe, AZ 85283, USA
kmitchel@uact.edu

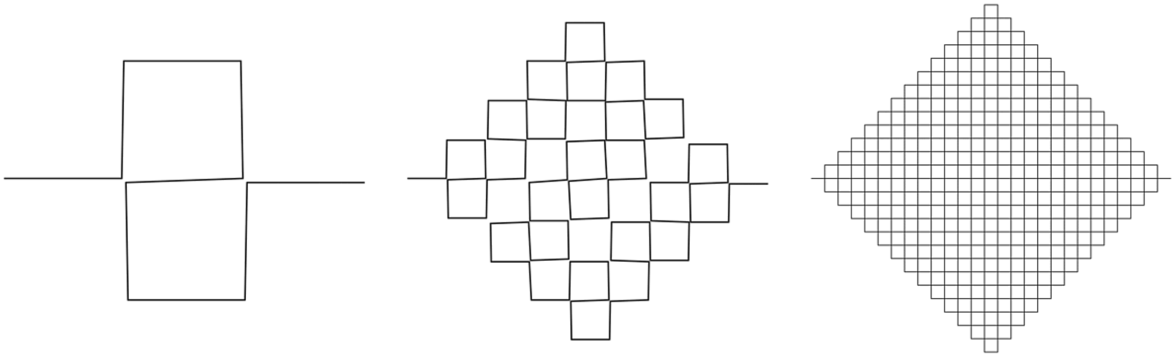
Abstract

Space-filling fractal curves are discussed, both classical and new. Techniques are discussed for rendering space-filling curves in an aesthetically pleasing fashion, in opposition to the traditional manner of black lines on white backgrounds. These techniques use color, line weight, and curved shapes, and can be applied to a wide variety of space-filling curves.

Introduction

Space-filling curves are those curves that, while having zero thickness, are sufficiently contorted that they completely cover an area or volume. Originally considered pathological, they have been part of the fractal family for over 100 years. In 1890, Giuseppe Peano first discussed the plane-filling curve [1], a fractal curve that literally filled a region of a plane. The next year, David Hilbert published the first image of a fractal, the two-dimensional square-filling curve bearing his name [2]. Since then, many other types of space-filling curves (as they are now called) have been discovered (or invented), such as Césaro's 1905 modification of the Koch curve [1].

Like other fractals, space-filling curves are generated in stages, or iterations. In Figure 1, the first few stages of the Peano curve are shown. The initial stage is a single line segment, and the generator is a nine-segment curve (the angles are actually 90°, but are shown as 89° for clarity). At each iteration, every segment in the current curve is replaced by a small copy of the generator.



Iteration 1: Generator

Iteration 2

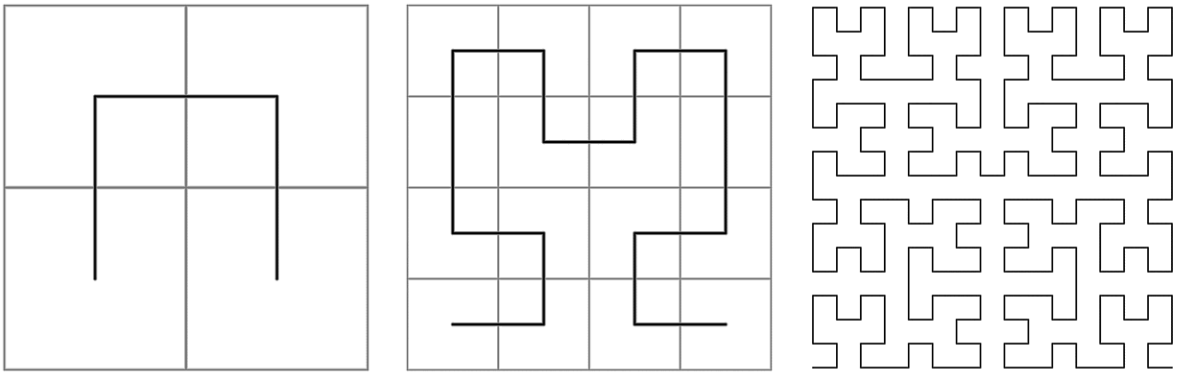
Iteration 3

Figure 1: Peano Curve. The angles in the first two panels have been reduced to 89° for clarity.

Typically, interest in these fractals has been academic, not artistic. Consequently, the curves have been rendered as in Figure 1, a series of black line segments on a white background. While appealing to the purist, this meager rendition does nothing to reveal the intrinsic beauty of such curves. This paper discusses some alternative techniques for rendering space-filling curves, resulting in images that are both mathematically substantive and aesthetically pleasing.

The Hilbert Curve

The Hilbert curve is a continuous curve that literally fills an entire square. Hilbert's generation of the curve began with a square divided into a 2x2 grid of smaller squares. On this grid, a three-segment curve is drawn, connecting the centers of the four small squares. This is the generator for the curve. The curve begins in the lower left corner, moves to the upper left, upper right, and then ends in the lower right corner. The first three stages of the generation are shown in Figure 2.



Iteration 1 with grid

Iteration 2 with grid

Iteration 4

Figure 2: Hilbert Curve

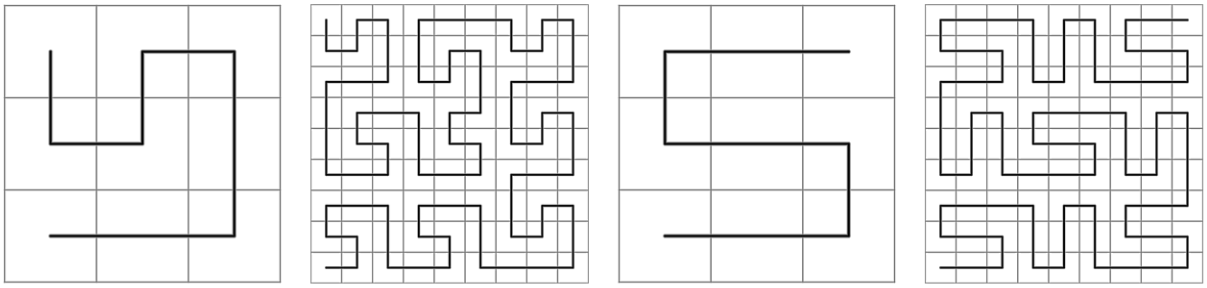
In the next stage of generation, each of the four squares is broken into a 2x2 grid of smaller squares. The curve from the first stage is shrunk to half its original size and copied four times: in the lower left quadrant, it is rotated clockwise a quarter turn, in the upper left and right quadrants, the curve is simply copied, and in the lower right quadrant, the curve is rotated a quarter turn counter-clockwise. The rotations are necessary so that the overall curve can begin in the overall lower left corner and end in the overall lower right corner. Between copies of the first stage curve, three small segments are added as connectors.

The rules are the same for each subsequent stage. Four copies of the previous curve are used, each one half of the previous size. In the upper left and right quadrants, the curve is copied unaltered. In the lower left quadrant, the curve is rotated a quarter turn clockwise, and a quarter turn counter-clockwise in the lower right quadrant. Three smaller connectors are used to join the copies. Each stage of the curve begins in the lower left corner, winds through the quadrants in the same order (lower left, upper left, upper right, lower right), and ends in the lower right corner. The curve is composed entirely of horizontal and vertical segments, it never crosses itself, nor does it ever touch itself.

Higher Order Curves

This geometric generation of the Hilbert curve used a decomposition of a square into a 2x2 grid of sub-squares. (There are other ways to generate the Hilbert curve, for example, using Lindemeyer systems [2].) Other space-filling curves can be generated using 3x3, 4x4, or nxn grids, by adhering to a few simple rules:

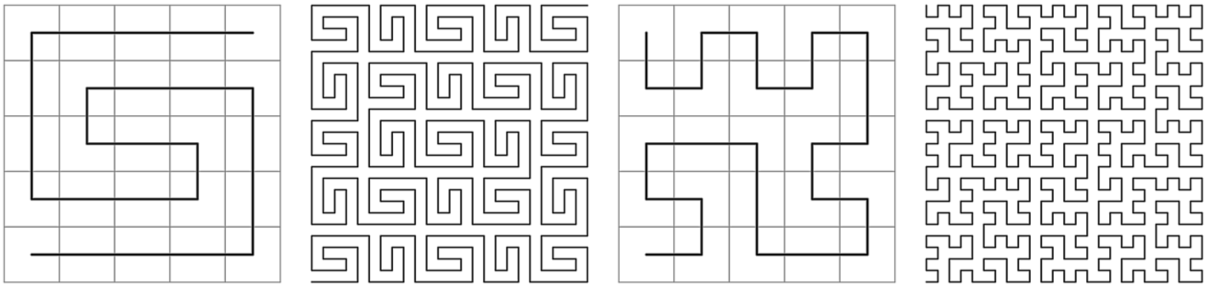
- The generator of the curve must pass through each square of an nxn grid.
- Each iteration of the curve must be composed of only small copies (rotated or flipped if necessary) of the generator, plus horizontal or vertical connectors between the copies.
- Each iteration of the curve must begin in one corner of the grid of squares and end in a different corner.
- The curve must not cross nor touch itself.



n31, 1 iteration n31, 2 iterations n32, 1 iteration n32, 2 iterations

Figure 3: Curves Based on 3x3 Generators

Up to rotation, the Hilbert curve is the only one possible from a 2x2 grid using these rules. On a 3x3 grid, there are two generators (up to rotation and flips), shown in Figure 3. One of the generators is an "S" curve, which is possible whenever n is odd. Somewhat arbitrarily, this generator is labeled "n32," and the other is "n31." With a 4x4 grid, there are 5 unique generators, one of which is just the second stage of the Hilbert curve. Two of the over 100 5x5 generators are shown in Figure 4. For clarity, the underlying grids have been removed in the "2 iteration" cases.



n5009, 1 iteration n5009, 2 iterations n5105, 1 iteration n5105, 2 iterations

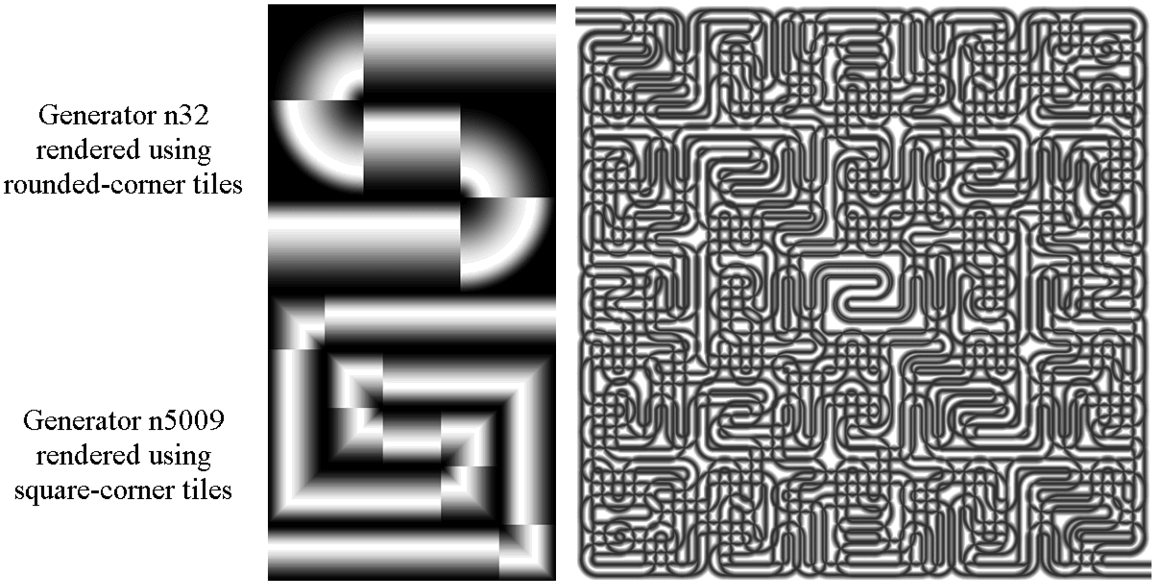
Figure 4: Curves Based on 5x5 Generators

Rendering Techniques

Taken to its logical conclusion, any space-filling curve would be rendered as a solid area. Consequently, the images illustrating these techniques generally use a low-iteration version of the respective curve, such that the sense of the curve can be determined, but there is still enough open space to view the rendering method.

Conceptually and operationally, an iteration of any of these curves can be considered to be defined by the sequence of its corner points. Thin segments can be drawn from one point to the next, giving the standard way the curves are shown.

Thick segments and Tiles. One way to add color is to thicken the line segments and color them. This was the basic approach used to create the image "Synchronicity," (by the author) shown in Figure 5. In practice, each corner point was represented by a tile indicating the direction of the curve through that point. Rounded or mitered corner tiles could be used, and the color was determined by the distance of a point on the tile from the center of the curve. In the figure, the coloring orientation of the tiles along the diagonal (top left to bottom right) has been reversed to emphasize the size and shape of the tiles.



Generator n32
rendered using
rounded-corner tiles

Generator n5009
rendered using
square-corner tiles

"Synchronicity"

Figure 5: Rendering Curves with Thick Lines and Tiles

The "synchronous" aspect comes from the fact that two separate curves were overlaid to create the final image. Both had "S" curve generators, one on a 3x3 grid and the other on a 5x5 grid. The third iteration of the 3x3 curve (27x27 squares) was combined with the second iteration of the 5x5 (25x25 squares), and the two curves "synchronized" in the center.

String Art. In the "string art" method, the image is rendered by drawing a series of colored lines from one set of points on the curve to another. For each point in the sequence, four control points are identified: A1, B_1, A_2, and B_2, being the endpoints of two control lines L_1 and L_2. A_1 and B_1 are the endpoints of line L_1 and A_2 and B_2 are the endpoints of line L_2. Along each control line, some number of points is established. Then, colored lines are drawn from the first point on line L_1 to the first point on line L_2, up through the last point on line L_1 to the last point on line L_2. The color of each line can be determined by the line's length, angle, or any other criterion. If the number of lines is relatively small, then an open mesh-type image will result. Using many lines will close the mesh and result in a solid mass of color.

Figure 6 shows a simple case where point A_1 is at (0,0) (lower left), B_1 and A_2 are at (1,0) (lower right), and B_2 is located at (1,1) (upper right). On the left, eleven solid black lines were used and the open mesh is quite apparent. On the right, 1100 lines were used, and their thickness adjusted such that the image appears to be a solid region. The lines were colored in accordance with the order in which they were drawn; this gives the appearance of the region being filled with a colored gradient.

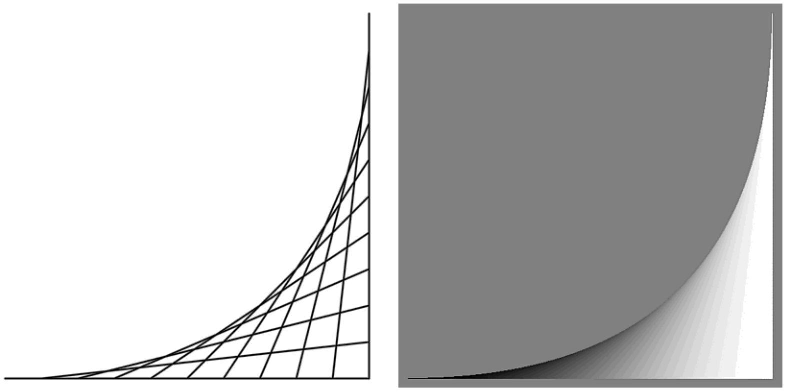
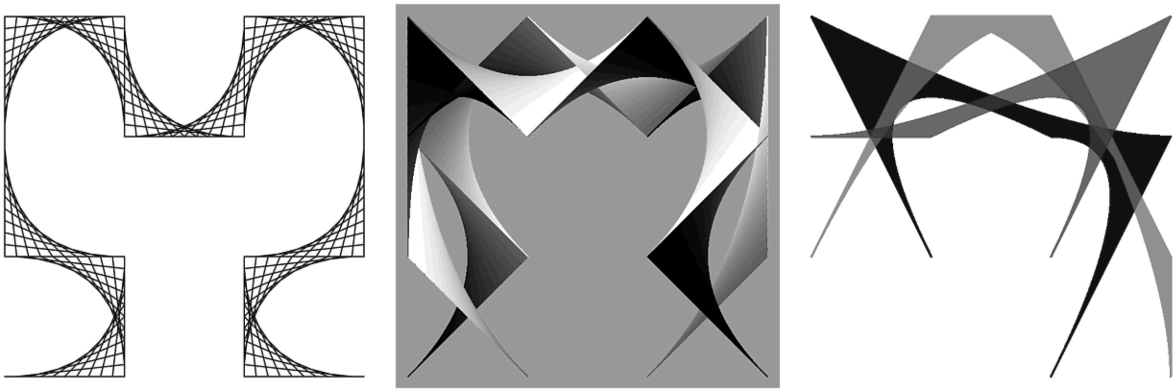


Figure 6: “String Art” Method

When applying this technique to the sequence of points in an iteration of a space-filling curve, one must decide how the four control points will be determined for each sequence point. Figure 7 was based on the second iteration of the Hilbert curve. On the left, the four control points were: the current point was point A_1, the next point was both B_1 and A_2, and the next point was the point B_2. This was represented more compactly by the set (0, 1, 1, 2), the four offsets from the current point to determine the control points. The resulting image had each of the corner regions replaced by the curve-mesh shape in Figure 6. In the center of Figure 7, more lines were used to give the appearance of solid mass. The control points were (0, 2, 2, 4). As the reach of the control points grows, the apparent iteration level of the curve decreases. However, the actual level can be seen in the details of the image.

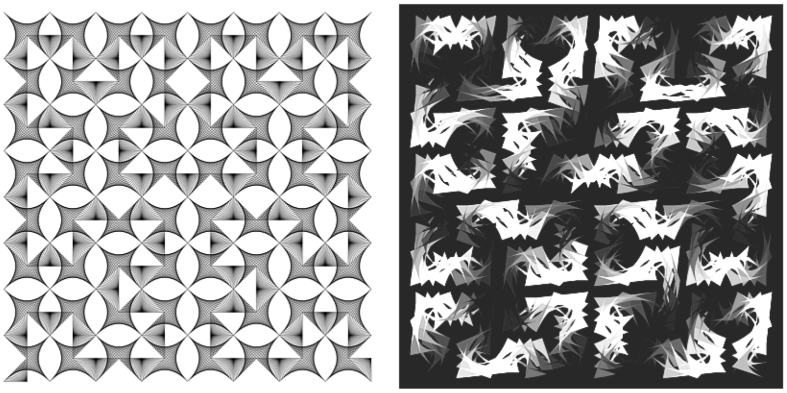


Offsets (0, 1, 1, 2) Offsets (0, 2, 2, 4) Offsets (0, 3, 3, 6)

Figure 7: “String Art” Method Applied to Second Iteration of the Hilbert Curve

Two other parameters determine which points in the sequence are used: the starting point and the increment. Depending on the control points used, the image may have structures that relate to every other point or every third point or some other periodicity. These structures can be illuminated by changing the increment. In right panel of Figure 7, the control points were (0, 3, 3, 6). Each curve used only every third sequence point of the second iteration of the Hilbert curve. The blue curve began with the first point in the sequence, the green began with the second point, and the red curve began with the third point.

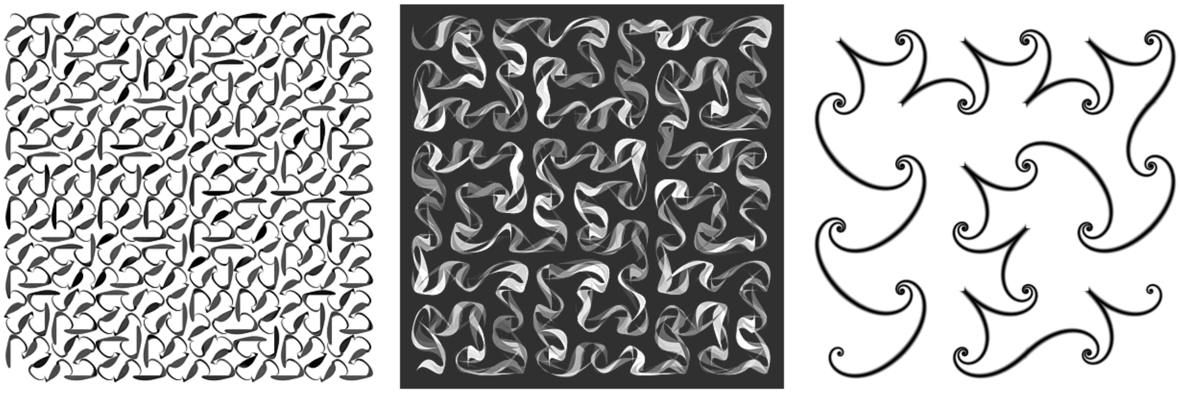
These effects came together in two images created by the author, "Hilbert's Ghost," and "Hilbert's Rainbow 2," shown in Figures 8. In the former, relatively few lines were used to leave the open feeling. Also, the sequence of the control points was chosen such that both the String Art curves and the original linear design are seen. In "Hilbert's Rainbow 2," a non-symmetric sequence of control points led to the off-axis lean of the image. Also, many lines and a gently varying color palette contributed to the three-dimensional effects. Both images used the fifth iteration of the Hilbert curve.



"Hilbert's Ghost" "Hilbert's Rainbow 2"

Figure 8: "String Art" Images Based on the Hilbert Curve

Rendering with Curves. The "string art" method can be extended by using other base curves instead of lines. Three samples, created by the author, are shown in Figure 9. For example, a parabola can be drawn between three points. Thus, nine points can be taken from the sequence for a given iteration and used to determine three control parabolas. Then, points along each control curve are used to determine the parabolas that are actually drawn to render the image. This method was used to create "Alhambra," based on the Hilbert curve. "Dance of the Veils" was rendered with parabolas and the third iteration of the n31 generator. Other curves, such as higher order polynomials, splines, or sinusoids can be used, once the basic method is understood.



"Alhambra" "Dance of the Veils" "Synergistic Spirals"

Figure 9: Rendering with Curves

"Synergistic Spirals" (named by Carlo Sequin [3]) was rendered using spirals. Each spiral segment was defined by two points, the pole about which the spiral wound, and the free end. In addition, the slope of the spiral and the free end could be specified and the tightness with which the spiral was wound. In this image, the sequence points were alternatively used as the pole and free ends and the slopes were assigned such that the curve would naturally flow from one pole to the next. The underlying curve is the second iteration of a 6x6 generator.

Conclusions

Space-filling curves represent one of the oldest areas of fractal geometry, yet have been relatively untouched artistically. Traditionally, such curves have been of interest only to the mathematics community, and their rendering has been functional, but lacking aesthetically.

A method was shown by which an infinity of space-filling curves can be generated, based on the square-decomposition idea used by Hilbert. This method emphasizes the sequence of points that typically comprise the corners of an iteration of the curve. By using these points as the bases for drawing with lines or other shapes, a wide variety of artistic renderings can be generated.

References

[1] Mandelbrot, B., *The Fractal Geometry of Nature*, W.H. Freeman and Company, 1983.

[2] L.Peitgen, H.-O., Jürgens, H., and Saupe, D., *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, 1992.

[3] Sequin, C., private communication, 2002.