

Plane-Filling Trails

Herman Haverkort

Universität Bonn, Germany

<http://herman.haverkort.net/>

cs.herman@haverkort.net

Abstract

The order in which plane-filling curves visit points in the plane can be exploited to design efficient algorithms. Typically, the curves are useful because they preserve locality: points that are close to each other along the curve tend to be close to each other in the plane, and vice versa. However, sketches of plane-filling curves do not show this well: they are hard to read on different levels of detail and it is hard to see how far apart points are along the curve. This paper presents a software tool to produce compelling visualisations that may give more insight in the structure of the curves.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases space-filling curve, plane-filling curve, spatial indexing

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.81

Category Media Exposition

Related Version A full version of this paper is available at <https://arxiv.org/abs/2003.12745>.

Supplementary Material For software and additional figures visit <http://spacefillingcurves.net>.

1 Plane-filling curves

A plane-filling curve is a continuous surjective mapping f from the unit interval to a subset of the plane that has positive area, that is, Jordan content. Although such a mapping cannot be one-to-one, an unambiguous inverse can be defined with a tie-breaking rule. Thus, the mapping provides an order in which to process points in the plane. Famous examples include Pólya's triangle-filling curve [12] and square-filling curves by Peano [11] and Hilbert [7]. Continuity of the mapping is not always required: if we drop this requirement, we speak of plane-filling *traversals*. Z-Order [9] is an example that is often applied in practice.

Plane-filling traversals and their inverses have been used to design efficient solutions for various applications, including indexing of points in the plane, geometric algorithms and data structures, finite element methods, load balancing in parallel computing, improving cache utilization in computations on large matrices or images, combinatorial optimization, image compression, information visualization, and sonification [1, 6]. It is therefore interesting to see the differences between the various plane-filling traversals that have been proposed.

2 Defining a plane-filling curve

Plane-filling traversals are usually visualised in a way that follows their definition. Consider Pólya's curve. To define it, we start with a single line segment (Figure 1a). We refine this simple drawing as follows. Let p and r be the end points of the original line segment. Imagine a circle with centre line pr and draw another point q halfway on the circle as we follow it clockwise from p to r . Erase the original line segment pr and replace it by two smaller segments pq and qr (Figure 1b). Next, refine the drawing again by applying the same refinement procedure to each segment, but this time changing the orientation: to find the new intermediate points, we now follow the circles in counterclockwise direction. To indicate this change in orientation, we add an arrow head to pr on the left side, and put the arrow heads for pq and qr on the right side. Thus, two line segments become four segments



© Herman Haverkort;

licensed under Creative Commons License CC-BY

36th International Symposium on Computational Geometry (SoCG 2020).

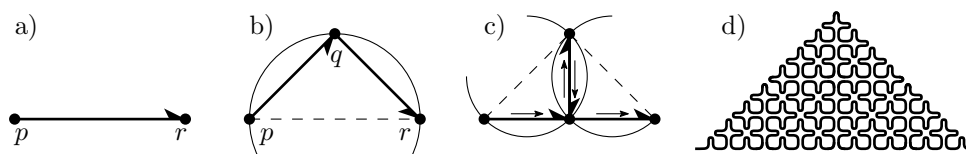
Editors: Sergio Cabello and Danny Z. Chen; Article No. 81; pp. 81:1–81:5

Leibniz International Proceedings in Informatics

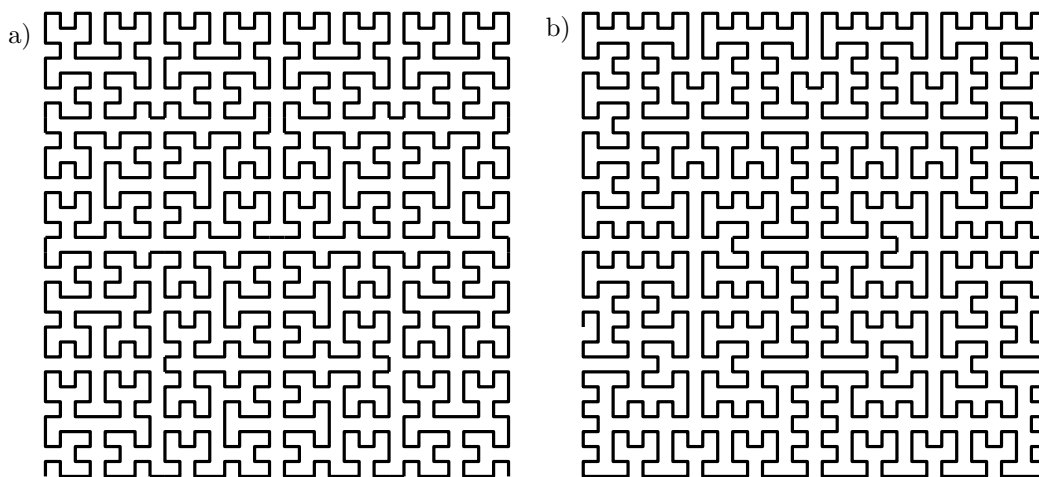


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Pólya's triangle-filling curve.



■ **Figure 2** Sketches of a) the Hilbert curve [7] and b) an Ω section of the $\beta\Omega$ -curve [16].

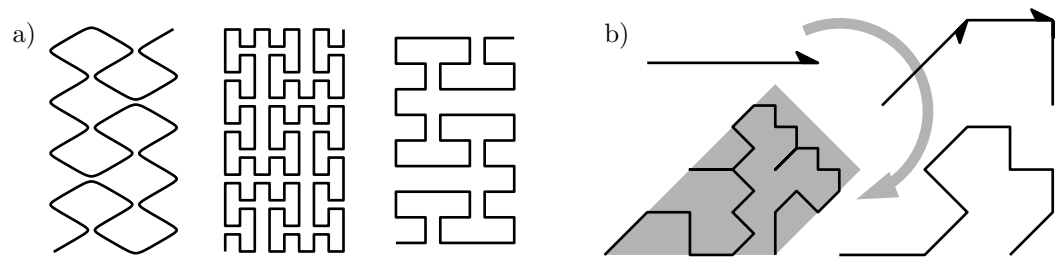
(Figure 1c). Note that the middle two segments lie on top of each other, but they have different directions. If we repeat this refinement process six more times, alternating clockwise and counterclockwise, and move all points slightly so that the curve does not back up on itself, we get Figure 1d. If we continue ad infinitum, the curve fills a right isosceles triangle.

3 The challenge of visualising plane-filling curves

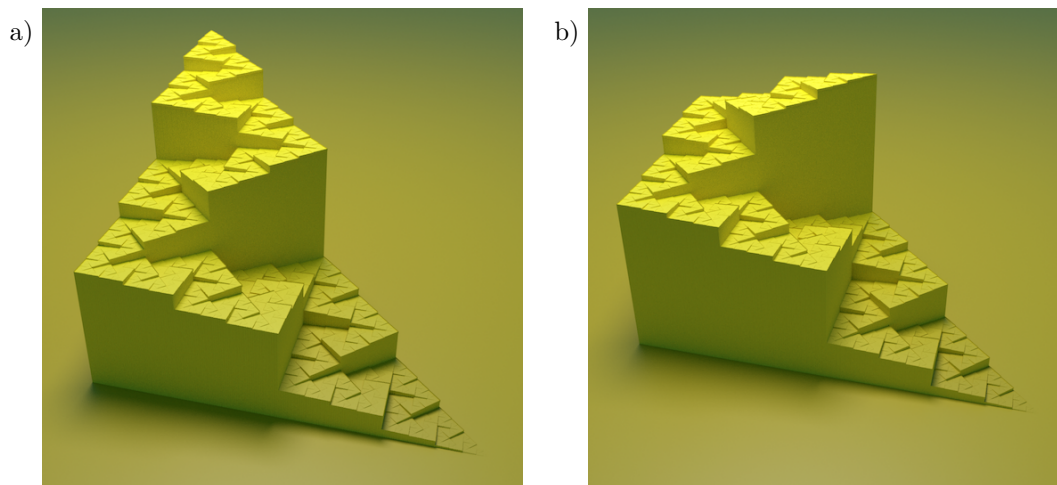
Figures 1b and 1d are typical of the way in which plane-filling curves are usually sketched. Neither figure makes it clear in an instant in what order the curve fills what parts of the plane – not to mention showing the curve's locality-preserving properties and violations thereof. Try comparing, for example, Hilbert's curve in Figure 2a to the $\beta\Omega$ -curve in Figure 2b (a promising alternative [17]). Furthermore, the impression one gets of the curve depends heavily on how one chooses to define it and on the details of how it is sketched. Figure 3a shows three sketches that all sketch the same curve, and Figure 3b shows a sketch of a trapezoid-filling curve that is nothing else than the first three quarters of Pólya's curve: none of this is visually obvious from the drawings.

4 Visualisation as three-dimensional landscapes

To visualise plane-filling curves and traversals more clearly, I developed a tool `pftrail`. The tool reads a definition of a plane-filling curve and produces a *plane-filling trail*, a model of the curve on a three-dimensional landscape, in which each point $f(t) = (x, y)$ of the curve is rendered as a point (x, y, t) . Thus the curve becomes a steadily ascending path in the landscape, see Figure 4. At a low resolution, the concept can be seen in action in a Hilbert curve marble run design by Ortiz [10]. At higher resolutions, we obtain a clear visualisation of the locality-preserving and locality-violating properties of the curve that can be studied at different levels of detail. High, steep slopes reveal pairs of points that are close in the



■ **Figure 3** a) Three sketches of Peano's curve, mapped to a $\sqrt{3} : 1$ rectangle. b) A sketch of a trapezoid-filling curve.

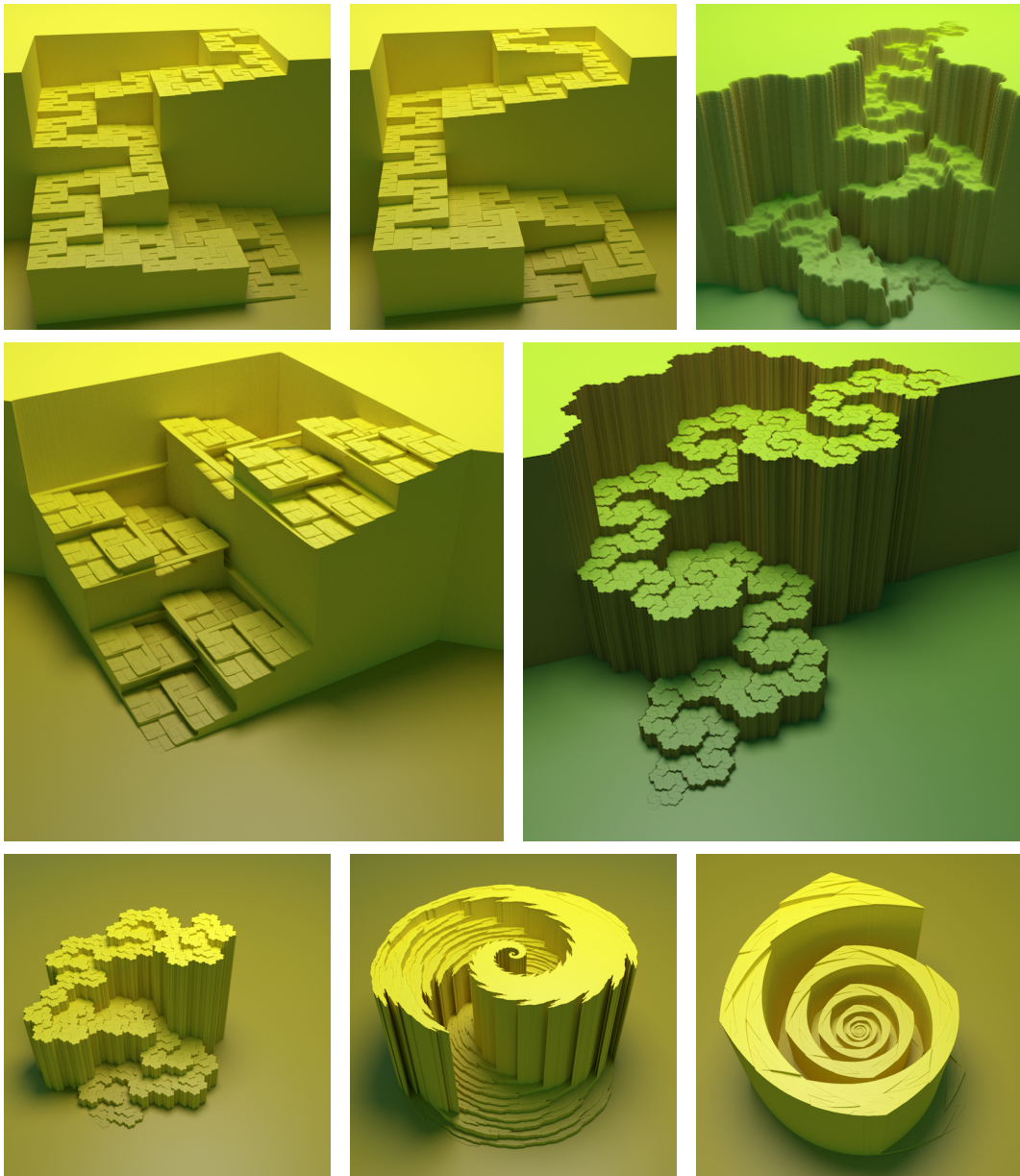


■ **Figure 4** a) Pólya's curve. b) The trapezoid-filling curve sketched in Figure 3b.

plane but far apart along the curve. Narrow corridors reveal sections between points that are relatively close to each other along the curve, but far apart in the plane. Wide corridors show sections of the curve that have good locality-preserving properties in both directions. The global course of the curve is easy to follow, but the image also facilitates studying the curve in more detail. Moreover, the visualisation is independent of what definition of the curve is used, out of multiple equivalent definitions. For example, the fact that the trapezoid-filling curve is simply the first three quarters of the Pólya curve is now obvious, see Figure 4. The visualisation gives the user the possibility to study the curve without any bias towards an arbitrary underlying tessellation.

5 Alternative visualisations

Alternative methods that come closest to meeting the same goals render the t -coordinate as values on a grey or colour scale instead of elevation. Indeed, such renderings are quite common. However, in comparison to our perception of elevation in a landscape, our perception of colour is less precise, more context-dependent, and not invariant under translation. Another interesting alternative are the three-dimensional models by Irving and Segerman [8] that stack different refinement levels according to a definition of the curve. However, these models are hard to “read” when presented as a two-dimensional printed image, and they are inherently dependent on the chosen definition of the curve. That is fine if one wants to illustrate the definition, but it is a shortcoming if one wants to be able to reveal the equivalence of different definitions by producing the same image in such cases.



■ **Figure 5** On top: Hilbert’s curve [7], $\beta\Omega$ [16], and a curve from Ventrella [15] (p86) filling a fractal “pinwheel” tile [2], rendered “eroded”. In the middle: Double-Gray-code [4] and a curve filling half of a Rauzy fractal [13]. Bottom: the Gosper curve [5], a close-up of the point at $2/7$ of the curve where three tiles meet ($\zeta = 8$), and a close-up of the point at $1/3$ of Pólya’s curve ($\zeta = 5$).

6 The pftail tool

The `pftail` tool reads the definition of a plane-filling traversal in the format from Ventrella [15], extended to support discontinuities and multiple refinement rules (known as *generators*). Thus, the various traversals that have been proposed in the computer science literature [3, 14, 16] can all be rendered and it is easy to explore new designs. Traversals are not confined to an integer grid, so we can also visualise interesting traversals related to, for example, the Rauzy

fractal [13] (see Figure 5). For rendering, the traversal is sampled and drawn on a grid of hexagonal cells; thus `pftrail` operates without any knowledge of the shape that is filled by the traversal (which can be a complicated fractal). The tool offers various options to control parameters such as camera position, resolution of the rendering grid, visualisation style, and the height of “parapets” that accentuate steep drops to enhance the perception of depth. The output is a `collada` file that can be rendered with, for example, Blender; if the resolution is not too high, it can also be moved around in Blender in real time.

Special features include “polynomial” close-up: given a focus point p and a zoom parameter ζ , any point q at distance x from p is moved to the point at distance $x^{1/\zeta}$ on the ray from p through q . Elevation differences are modified in a similar way. This allows us to zoom in on features that remain invisible in normal close-up views. For example, the Gosper curve (Figure 5, bottom left) follows a tessellation with tiles arranged in a hexagonal grid pattern. At the vertices of this grid, the tiles wind around each other like logarithmic spirals that shrink by a factor of roughly $9 \cdot 10^7$ per revolution. No normal close-up view could show these spirals, but the polynomial close-up reveals them clearly, see Figure 5.

References

- 1 M. Bader. *Space-filling curves: an introduction with applications in scientific computing*. Springer, 2013.
- 2 C. Bandt, D. Mekhontsev, and A. Tetenov. A single fractal pinwheel tile. *Proc. Amer. Math. Soc.*, 146:1271–1285, 2018.
- 3 C. Burstedde and J. Holke. A tetrahedral space-filling curve for nonconforming adaptive meshes. *SIAM J. Scientific Computing*, 38(5), 2016.
- 4 C. Faloutsos. Multiattribute hashing using Gray codes. In *Proc. 1986 Conf. ACM SIG Management of Data (SIGMOD 1986)*, pages 227–238, 1986.
- 5 M. Gardner. Mathematical games: in which “monster” curves force redefinition of the word “curve”. *Scientific American*, 235:124–133, 1976.
- 6 H. Haverkort. Sixteen space-filling curves and traversals for d -dimensional cubes and simplices. *CoRR*, abs/1711.04473, 2017. [arXiv:1711.04473](https://arxiv.org/abs/1711.04473).
- 7 D. Hilbert. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.*, 38(3):459–460, 1891.
- 8 G. Irving and H. Segerman. Developing fractal curves. *J. of Mathematics and the Arts*, 7(3–4):103–121, 2013.
- 9 G. Morton. A computer oriented geodetic data base, and a new technique in file sequencing. Technical report, International Business Machines Co., Ottawa, Canada, 1966.
- 10 S. Ortiz. Hilbert curve marble run, 2018. Accessed 26 March 2020. URL: <https://www.thingiverse.com/thing:3031891>.
- 11 G. Peano. Sur une courbe, qui remplit toute une aire plane. *Math. Ann.*, 36(1):157–160, 1890.
- 12 G. Pólya. Über eine Peanosche Kurve. *Bull. Int. Acad. Sci. Cracovie, Ser. A*, pages 305–313, 1913.
- 13 G. Rauzy. Nombres algébriques et substitutions. *Bulletin Soc. Math. Fr.*, 110:147–178, 1982.
- 14 H. Samet. *Foundations of multidimensional and metric data structures*, page 199. Morgan Kaufmann, 2006.
- 15 J. Ventrella. *Brainfilling curves: a fractal bestiary*. Eyebrian books, 2012.
- 16 J.-M. Wierum. Definition of a new circular space-filling curve: $\beta\Omega$ -indexing. Technical Report TR-001-02, Paderborn Center for Parallel Computing (PC²), 2002.
- 17 S.-E. Yoon and P. Lindstrom. Mesh layouts for block-based caches. *IEEE Trans. on Visualization and Computer Graphics*, 12(5), 2006.