Communication Papers of the 18th Conference on Computer Science and Intelligence Systems pp. 259–264
DOI: 10.15439/2023F5773
ISSN 2300-5963 ACSIS, Vol. 37

A simple algorithm for computing a multi-dimensional the Sierpiński space-filling curve generalization.

Ewa Skubalska-Rafajłowicz, Member, IEEE
0000-0002-2795-3835
Dept. of Computer Engineering,
Wroclaw University of Science and Technology,
Wyb. Wyspianskiego 27, 50 370 Wrocław, Poland,
Email: ewa.skubalska-rafajlowicz@pwr.edu.pl

***Abstract*—Transforming multidimensional data into a one-dimensional sequence using space-filling curves, such as the Hilbert curve, has been studied extensively in many papers. This work provides a systematic presentation of the construction of an arbitrarily accurate multidimensional space-filling curve approximation which is a generalization of the Sierpiński space-filling curve. At the same time, according to the space-filling curve construction, we present a simple algorithm for determining one of the counter-images on a unit interval of a data point lying in a multidimensional cube. The computational complexity of the algorithm depends linearly on the dimension of the cube. The paper contains numerical algorithms for local generation of the curve approximation and determination of the quasi-inverse of a data point used to transform multidimensional data into the one-dimensional form.**

I. Introduction

THE space-filling curve (SFC) is defined as a continuous mapping converting a unit interval $[0,1]$ onto the d-dimensional unit cube ($I_d = [0,1] \times \ldots \times [0,1]$, $d \leq \infty$) [27], [2]. This means that the space-filling curve passes at least once through each point of the I_d cube. This allows many multidimensional computational problems to be considered as one-dimensional problems without losing the essential properties of the original problems.

Space-filling curves were first described by G. Peano in 1890 [25], and then by D. Hilbert [12], and W. Sierpiński [30].

In the 1930s, space-filling curves, which are measure preserving, have been used in the theory of integration in multidimensional spaces [27], [18].

The applications of space-filling curves are pretty broad, though in many cases, they are limited to two-dimensional curves. We can mention as examples: image processing [24], image compression [22], image encryption [4], image malware classification [23], MRI image sampling [29], the cryptographic transformation scheme for spatial query processing [13], encryption technology for data privacy-preserving [16], raster tool path generation for layered manufacturing [14], among many others. Nair et al. [21] used the Hilbert space-filling curve to explore the space of the robot and detect obstacles.

The Sierpiński space-filling curve was applied to solve discrete optimization problems [41]. In particular, Bartholdi and Platzman [3], [26] applied the Sierpiński curve to find an approximate, near optimal solutions of the planar traveling salesman problem with Euclidean distances. An important property of a space-filling curve is the preservation of proximity by the points from $[0,1]^d$ transposed to $[0,1]$. This property means that points lying close to each other on the curve are also close in the multidimensional space. Points far apart on a curve can be close to each other in the multidimensional space. In order to estimate the actual distance of a pair of points in $[0,1]^d$, say (x,y) one should have determined all $2*2^d$ points on the unit interval whose images are respectively x and y. Hence the idea of using space-filling curves to find nearest neighbors of subsequent point from multi-dimensional space.

Multidimensional space-filling curves are also used in global optimization algorithms [31],[32], [33] and their applications, e.g. in experimental design [35]. Lawder et al. [15] discuss multidimensional indexing for database management systems based on space-filling curves.

On the other hand, the space-filling-based transformations retain essential statistical information. For example, it is proved that the Bayes risk is invariant under these transformations for every distribution with a bounded support [38].

Sampling of multidimensional space using one-dimensional equidistributed sequences transformed by a multi-dimensional space-filling curve was developed in [40], [11]. Data-dependent space-filling curves for non-uniform grid were proposed by[34],[43].

The Hilbert space-filling curve was used in parallel codes for numerical simulations [5], and the Sierpiński curve helped to streamline finite element calculations [1]. and for load-balancing in distributed computing [17].

The algorithms in which space-filling curves were used to reduce the dimension of data and then to analyze them concerned, among others, the determination of the box-counting fractal dimension [39], the classification of multidimensional data [38] and the data clustering [19], [42].

It is known, that the continuous mapping $F_d : I_1 \to I_d$

can not be one-to-one [27], [3], [26]. The geometric points of intersection of the curve with itself correspond to many points of the unit interval. Thus, from the viewpoint of such possible applications, it is crucial to find $t \in I_1$ such that $F_d(t) = x$ for given $x \in I_d$, i.e., to provide a quasi-inverse of $F_d(t)$.

This paper provides a fast and relatively simple algorithm for computing the approximate quasi-inverse for any dimension d. Naturally, this is intrinsic to the definition of the space-filling curve generation. One can transform each element of the multidimensional data set using quasi-inverse separately and at any moment. It does not require the construction of the entire space-filling curve. Such transformation allows us to have a linear order of data in higher dimensions. Since the curve is a closed curve, the resulting order is cyclical.

The outline of the paper is as follows. Section 2 introduces the main ideas of constructing the d-dimensional Sierpiński space-filling curve generalization. Section 3 contains two algorithms. Algorithm 1. implements the construction of the nodal point of the d-dimensional Sierpiński curve, i.e., it transforms the selected data point from the unit interval onto the d-dimensional unit cube. Algorithm 2 implements the quasi-inverse mapping connected to the generalized Sierpiński curve, i.e., it allows us to obtain a one from 2^d positions on the unit interval of a given point from the d-dimensional hypercube. The example of using the algorithm to transform 4-dimensional Iris data into unit interval is given at the end of the section. The last section summarizes the contents of the paper.

II. The method of constructing the d-dimensional space-filling curve.

The method of constructing the d-dimensional space-filling curve can be related to the sequential division of the filled multidimensional space into elementary areas, usually multidimensional sub-cubes of the same shape and the same volume [20].

Next, a one-to-one correspondence is established between the 2^{dk} elementary intervals U_k of the length 2^{-dk} and between the $(2^d)^k$ sub-cubes C_k of size $2^{-k} \times 2^{-k} \ldots \times 2^{-k}$ ($k = 1, 2, 3, \cdots$). The correspondence is such that any two adjacent sub-intervals correspond to two adjacent sub-cubes and moreover, 2^d sub-intervals U_{k+1} (of the length $2^{-d(k+1)}$) which constitute a sub-interval U_k correspond to the 2^d sub-cubes C_{k+1} associated with the corresponding sub-cube C_k. In this correspondence, adjacent sub-cubes of any level of division k are related to adjacent subintervals of the same degree of partition in the unit cube [20], [18], [27]. In this way, one can define the classical space-filling curves such as the Peano, Hilbert, and Sierpi ́nski.

Define the family W of 2^d mappings $w_i : R^d \to R^d,\ i = 0, \ldots, 2^d - 1$ of the following form:

$$w_i(x_1, x_2, \ldots, x_d) = \begin{cases} \frac{1}{2} - (\frac{1}{2} - \beta_{1,i})x_1 \\ \frac{1}{2} - (\frac{1}{2} - \beta_{2,i})x_2 \\ \cdots \\ \frac{1}{2} - (\frac{1}{2} - \beta_{d,i})x_d \end{cases} \quad (1)$$

where $\beta_{j,i} \in \{0, 1\},\ \ j = 1, 2, \ldots, d,\ \ i = 0, 1, \ldots, 2^d - 1$. For $\beta_{j,i} = 0$ we get the transformation of the form $\frac{1}{2} - \frac{1}{2}x_j$ and for $\beta_{j,i} = 1$ we get $\frac{1}{2} + \frac{1}{2}x_j$. Mappings w_i are indexed in such a way that vectors $B_i^d = (\beta_{1,i}, \beta_{2,i}, \ldots, \beta_{d,i})^T$ determine w_i uniquely.

B_i^d forms a list of d–dimensional vectors containing only 0 and 1, where each pair of the adjacent vectors differs at exactly one position. In geometric terms, such a list of vectors describes a closed (Hamiltonian) path through all vertices of d-dimensional unit cube ($I_d = [0, 1] \times [0, 1] \times \ldots \times [0, 1]$).

Among the many different possibilities, we limit ourselves here to the order defined by the classical, reflective (reflected) binary Gray code (see, e.g., [9]). $d + 1$–dimensional code is formed from the d–dimensional one as follows:

$$B_0^{d+1}, \ldots, B_{2^d-1}^{d+1} = (B_0^d, 0), \ldots, (B_{2^d-1}^d, 0)$$

and the reverse order added

$$(B_{2^d-1}^d, 1), \ldots, (B_0^d, 1)$$

For example, for $d = 3$ we obtain a closed sequence of the vertices of the 3 dimensional unit cube in the following form closely related to the sequence of mappings $(w_0, w_1, \ldots, w_7)$ and $w_8 = w_0$:

$$(000), (001), (011), (010), (110), (111), (101), (100), (000).$$

Define a number sequence $b_k,\ k = 1, \ldots$ such that:
$b_1 = 1\ ,\ b_k = 2^{k-1} - b_{k-1} + 1,\ k = 2, 3, \ldots$

In this way, we obtain a fast-growing sequence of the positive integers $(1, 2, 3, 6, 11, 22, \ldots)$, index sequences that specify the position of vertex $(1, \ldots, 1)$ in the sequence (and the corresponding mappings) scanning the vertices of I_d.

Furthermore, it is easy to verify that:

Property 1: $b_d \to \infty$ as $d \to \infty$, but $2^{-d}b_d$ ranges from $\frac{1}{2}$ to $\frac{1}{3}$ as d ranges from 2 to infinity.

The number b_d is equal to the smallest distance between two of the most distanced vertices of the cube: vertex $(1, 1, \ldots, 1)$ and vertex $(0, 0, \ldots, 0)$ and as a consequence

$$B_{2^d-b_d}^d = (1, 1, \ldots, 1).$$

$w_{2^d-b_d}$ maps vertex $(1, 1, \ldots, 1)^T$ to the same vertex $(1, 1, \ldots, 1)^T$.

The mappings w_i show how the unit cube is split onto 2^d smallest sub-cubes (of size 2^{-1},$2^{-2}\ldots$) . Successive repetition of such partitions produces a sequence of $(2^d)^k$ sub-cubes, which were obtained by consecutive mappings with indices differing by one, i.e., defined by repeated sequences of numbers $s = (0, 1, 2, \ldots 2^d - 1, 2^d)$, where 0 and 2^d symbolize two parts of the same sub-cube (associated with the vertex corresponding to the starting point node $(0, \ldots, 0)$ of the current sub-cube). At each subsequent split, the sequence $(0, 1, 2, \ldots, 2^d - 1, 2^d)$ is replaced in the following way:

$$0 \to (2^d - b_d, \ldots, 2^d - 1, 2^d),$$

$$1 \to (0, 1, 2, \ldots, 2^d - 1, 2^d),$$

$$\ldots$$

$$2^d - 1 \to (0, 1, 2, \ldots, 2^d - 1, 2^d),$$
$$2^d \to (0, 1, \ldots, 2^d - b_d - 1, 2^d - b_d).$$

Notably, a space-filling curve is defined as a limit of the uniformly convergent space-filling curve approximations formed by line segments with ending points in adjacent sub-cubes. The approximations could differ, but the limit curve depends only on U_k and C_k structures [18], [27]. In our case, the endpoints lie on the chosen vertices of the subsequent sub-cubes. They have been chosen, so their positions do not change in subsequent iterations (for the next k). The next approximating curve is created by adding successive points without changing the location of the previous ones. It is worth emphasizing here that the refinement of the curve approximation takes place locally, separately in each sub-interval and the corresponding sub-cube. Refining the curve approximation in one of its fragments (a given sub-cube) does not affect refining the curve in its other fragment. The first partition of the unit interval consists of $2^d + 1$ sub-intervals with all intervals of the length $1/2^d$. The only exceptions were the first and last intervals, which are $b_d/2^{2d}$ and $(2^d - b_d)/2^{2d}$ (together add up to $1/2^d$). These two shorter sub-intervals correspond to the sub-cube with the vertex $(0, \ldots, 0)$, where the conventional beginning and end of the curve locate. The position of a point inside a particular sub-cube C_k indicates the sub-interval U_k where its counter-images find.

Transformations 1 are repeated without changing the scale of the cube, because the coordinates in subsequent divisions (in smaller and smaller scales) are each time scaled to the size of the unit cube by inverse transformations:

$$x_i = \begin{cases} 1 - 2x_i, \text{ if } x_i < 1/2, \\ 2x_i - 1, \text{ if } x_i \geq 1/2, \end{cases} \quad i = 1, \ldots, d. \tag{2}$$

Another possibility is to assume that $1 - 2x_i$ is performed when $x_i \leq 1/2$ and complementarily, transformation $2x_i - 1$ is performed for $x_i > 1/2$. Thus, the point with a finite binary expansion of all its coordinates can be connected to one of 2^d adjacent sub-cubes. Each such combination of partitions (for every coordinate, we can have partition $[0, 1/2), [1/2, 1]$ or $[0, 1/2], (1/2, 1]$) is enough to determine successive approximations of the quasi-inverse of the space-curve (see Algorithm 2), because the scaling of the length of the respective subintervals (by multiplying by 2^{-di}, $i \leq k$) is independent of the orientation of the currently considered sub-cube of side size 2^{-i}. Let's note that each point of the cube I_d is within $2^{-kd/2-1}$ distance from one of the vertices of the approximating curve.

Usually, the 2-D Sierpiński curve is defined as a correspondence between intervals and triangles. We will here rely on the construction of the 2-D Siepiński curve, which basis on a quadruple partition of a square (see Fig. 1), i.e., squares of side size 2^{-k}, where k is the number of the subsequent divisions of the unit square. As in proposed in this paper approach, the $b_d/2^{2d}$ and $(2^d - b_d)/2^{2d}$ intervals correspond to partitioning the cube onto two triangles in two dimensional space. When d is larger, the division of the multi-cube is more complicated, and the ratio of the volumes of the two parts depends on d.

The other version of the Sierpiński space-filling curve generalization can be obtained by changing the direction of passing the cube vertices in the sub-cubes obtained as a result of the w_i transformation with the odd index i.

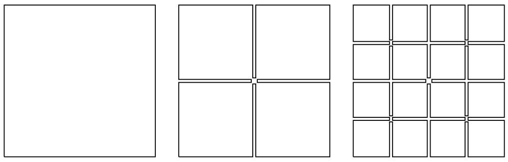

Fig. 1: Approximations of the Sierpiński space-filling curve in 2-D.

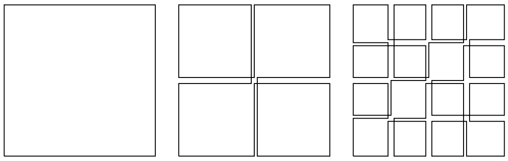

Fig. 2: Modification of the Sierpiński space-filling curve in 2-D.

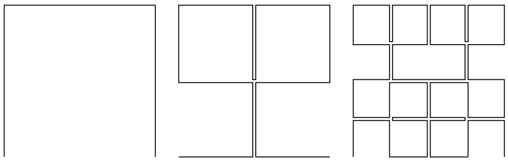

Fig. 3: Approximation of the Hilbert space-filling curve on the plane.

As a consequence, in the two-dimensional case, we obtain the agreement with the original Sierpiński curve (Fig. 1 in contrast to the 2-D space-filling curve roughly visualized in Fig. 2. Fig. 3 shows an approximation of the Hilbert curve. The Hilbert curve uses the same vertex order in the elementary cube as in the case of the Sierpiński curve [6], however, it does not form a closed cycle as in our case.

A. *Properties of the proposed family of the space-filling curves*

The previously defined sequence of approximating curves is uniformly convergent to the space-filling curve [20], [18].

The presented here generalization of the Sierpiński SFC treated as the map $F_d : I_1 \to I_d$, has the following properties:

a) $F_d(t)$ is a measure preserving map of I_1 onto I_d,
b) mapping F_d forms closed curve, i.e. $F_d(0) = F_d(1)$,
c) $F_d(t)$ is a Hölder continuous mapping of order $1/d$, in the following sense

$$\| F_d(t_1) - F_d(t_2) \| \quad \leq 2(d+3)^{1/2} \left(\Delta(t_1, t_2)\right)^{\frac{1}{d}}, \tag{3}$$

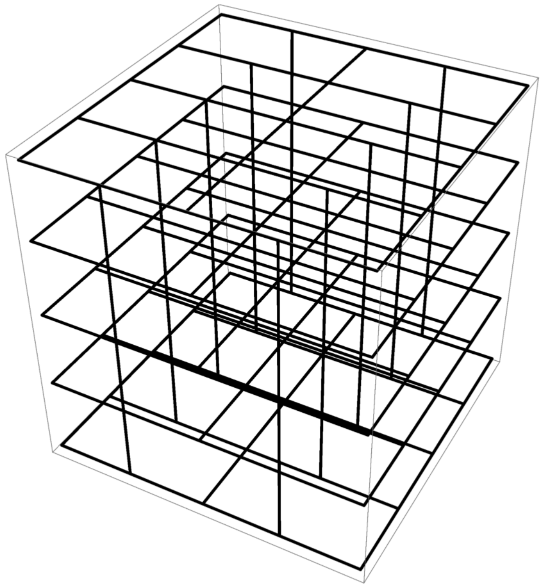

Fig. 4: Approximation of the proposed 3-D Sierpiński space-filling curve with 512 nodal points.

where $\Delta(t_1, t_2) = min\{\mid t_1 - t_2 \mid, 1- \mid t_1 - t_2 \mid\}$ for $t_1, t_2 \in [0,1]$is a metric on a circle, while $\| \cdot \|$ denotes the Euclidean norm in R^d. The constant $2(d+3)^{1/2}$ (in property c) is an upperbound. The same bound is valid for the multidimensional Hilbert curve [37], [11]. Furthermore, it is known that for the 2-D Sierpiński curve, the smallest possible constant equals 2 [26] or is close to $6^{1/2}$ for the 2-D Hilbert curve [10], [37].

III. The numerical algorithms

The first algorithm (Algorithm 1)approximates the points of the d-dimensional Sierpiński curve using the entered points from the unit interval. More precisely, it computes the image of any point $t \in [0,1]$ in $[0,1]^d$. The algorithm requires a number of iterations depending on the level of the curve approximation k, and the accuracy of the approximation of each coordinate point is 2^{-k}. The computational complexity of the algorithm is $O(kd)$. Figure 4 depicts an approximation of the 3-D Sierpiński space-filling curve.

The second algorithm (Algorithm 2) transforms a multivariate data point into a unit interval. The algorithm is iterative, and gives an approximation of the quasi-inverse of the d-dimensional Sierpiński curve, depending on the level of approximation, say k. The accuracy of determining the position on a unit interval of an image $x \in [0,1]^d$ is 2^{-dk}. The computational complexity of the algorithm is $O(kd)$.

In the case of two dimensions, the algorithms provide the curve (and its quasi-inverse) depicted on Fig. 1 b). A slight modification of the algorithms in places marked $\star$ allows us to obtain approximations of the original Sierpiński curve and its quasi-inverses.

The Fig. 5 illustrates the application of the transformation multivariate data on the example of Iris data [8], which contains 150 measurements (4 dimensional) of three different species of irises: Iris Setosa , Iris Versicolour, and Iris Virginica. It is easy to see that the Iris Setosa forms a separate cluster in the unit interval and only a small fraction of Iris Virginica is mixed with Iris Versicolor. Computations performed to generate Fig. 4 and Fig. 5 were made in Matematica 11.3 on the Intel(R) Core(TM) i7-6500U CPU, 2.50GHz.

Calculation of the transformation of a single point from a space with dimension $d = 4$ and accuracy 2^{-k}, $k = 10$ required approx. 0.000625 s. With the same accuracy and $d =$

Data: d, k, t, $(t \in [0,1])$
Result: $x \in [0,1]^d$
$x \leftarrow (1,1,\ldots,1)$;
$b_d \leftarrow 1$;
for $i \leftarrow 1$ **to** $d-1$ **do**
  $b_d \leftarrow 2^i - b_d + 1$;
end
$cd \leftarrow b_d * 2^{-d}$;
$KM \leftarrow [\,]$;
for $j \leftarrow 1$ **to** k **do**
  $km \leftarrow \lfloor t * 2^d - 1 + cd \rfloor + 1$;
  $(\star) t \leftarrow t * 2^d + cd - km$;
  if $km == 2^d$ **then**
    $km \leftarrow 0$;
  end
  append km to KM ;
end
for $j \leftarrow 1$ **to** k **do**
  $km \leftarrow KM[k-j+1]$;
  $B \leftarrow [\,]$;
  while $i < d+1$ **do**
    $be \leftarrow 1$;
    if $km < 2^{d-i}$ **then**
      $be \leftarrow 0$;
    end
    append be to B ;
    $km \leftarrow km - be * 2^{d-i}$;
    if $be == 1$ **then**
      $km \leftarrow 2^{d-i} - km - 1$;
    end
    $i \leftarrow i+1$;
  end
  for $i \leftarrow 1$ **to** d **do**
    $x_{d-i+1} \leftarrow 1/2 - (1/2 - B[i]) x_{d-i+1}$;
  end
end
modification for $d = 2$**:**
$(\star) t \leftarrow (cd + km - 1 + t)/2^d$ **by**
if km *is odd* **then**
  $t \leftarrow (-cd + km + 1 - t)/2^d$;
else
  $t \leftarrow (cd + km - 1 + t)/2^d$;
end

Algorithm 1: Mapping of $t \in [0,1]$ into a point $x \in [0,1]^d$

Data: d, k, $(x_1, x_2, \ldots, x_d)$, ($x \in [0,1]^d$)
Result: $t \in [0,1]$
$b_d \leftarrow 1$;
for $i \leftarrow 1$ **to** $d-1$ **do**
  $b_d \leftarrow 2^i - b_d + 1$;
end
$cd \leftarrow b_d * 2^{-d}$, $t \leftarrow 1 - cd$, $KM \leftarrow [\,]$, $B \leftarrow [\,]$;
for $j \leftarrow 1$ **to** k **do**
  for $i \leftarrow 1$ **to** d **do**
    if $x_i < 0.5$ **then**
      $be \leftarrow 0$, $x_i \leftarrow 1 - 2 * x_i$, **append** be to B ;
    else
      if $x_i \geq 0.5$ **then**
        $be \leftarrow 1$, $x_i \leftarrow 2 * x_i - 1$, **append** be to B ;
      end
    end
    $ww \leftarrow 0$, $km \leftarrow 0$;
    for $i \leftarrow 1$ **to** d **do**
      if $be + ww == 1$ **then**
        $km \leftarrow km + 2^{d-i}$;
      end
      $ww \leftarrow |be - ww|$;
    end
  end
  if $km == 2^d$ **then**
    $km \leftarrow 0$;
  end
  append km to KM;
end
for $j \leftarrow 1$ **to** k **do**
  $km \leftarrow KM[k - j + 1]$;
  $(\star) t \leftarrow (cd + km - 1 + t)/2^d$;
  if $t < 0$ **then**
    $t \leftarrow 1 + t$;
  end
end
modification for $d = 2$**:**
$(\star) t \leftarrow t * 2^d + cd - km$ **by**
$t \leftarrow t * 2^d + cd - km$;
if km *is odd* **then**
  $t \leftarrow 1 - t$;
end

Algorithm 2: Mapping of $x \in [0,1]^d$ into $t \in [0,1]$

2 the execution time was shortened twice.

IV. Concluding Comments

We present a simple algorithm for computing a transformation of multidimensional data points onto the unit interval using the proposed a Sierpiński type space-filling curve generalization. Specific ideas regarding the generalization of the Sierpiński curve were considered in the author's monograph [37], but the algorithms presented in this work are new and have not been published anywhere.

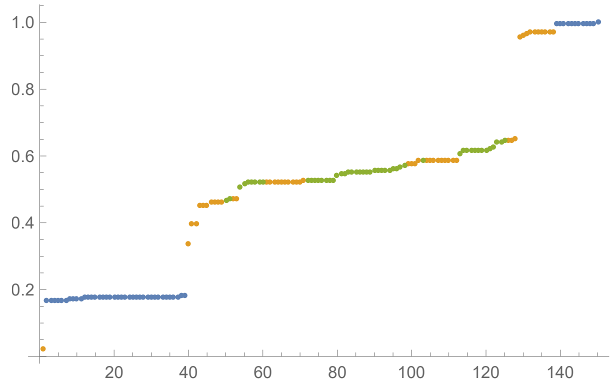

Fig. 5: Three species of the 4-dimensional Iris data: Iris Setosa (blue), Iris Versicolour (yellow), and Iris Virginica (green)- 150 data points - after dimensionality reduction.

It is known that there is a close relationship between topological dimension d of the I_d cube and the maximum value of the Hölder's exponent of a space-filling curve. The value of $1/d$ is the maximum value. There is no d-dimensional space-filling curve with an exponent greater than $1/d$ [18]. Hölder's inequality results in an important property of space-filling based transformations, which is to ensure the proximity of data points whose counter-images are closely located on a unit interval. Multi-dimensional Sierpiński curves are another tool for analyzing multidimensional data.

References

[1] Bader M., Schraufstetter S., Vigh C., Behrens J.: Memory effcient adaptive mesh generation and implementation of multigrid algorithms using Sierpinski curves, International Journal of Computational Science and Engineering, 4(1), 1742–7193, (2008).
[2] Bader M.: Space-Filling Curves. An Introduction with Applications in Scientific Computing. Springer-Verlag Berlin Heidelberg (2013).
[3] Bartholdi J. J., Platzman L. K.: Heuristics based on spacefilling curves for combinatorial problems in Euclidean space, Management Sci., 34, 291–305, (1988).
[4] G. Bhatnagar, Q. M. J. Wu, and B. Raman: Image and video encryption based on dual space-filling curves, Computing Journal,55(6), 667–685, (2012).
[5] Bungartz H.-J., M. Mehl M., T. Neckel T., and T. Weinzierl T.: The PDE framework Peano applied to fluid dy- namics: an efficient implementation of a parallel multiscale fluid dynamics solver on octree-like adaptive Cartesian grids, Computational Mechanics, 46(1), 103–114, (2010).
[6] Butz A.R.: Alternative algorithm for Hilbert's space-filling curve, IEEE Trans. on Computing, 20, 424–426 (1971).
[7] Falconer K.: , Fractal Geometry: mathematical foundations and applications, John Wiley & Sons Ltd., Chichester, 3-nd ed.(2014).
[8] Fisher,R.A.: The use of multiple measurements in taxonomic problems, Annual Eugenics, 7 II, 179–188 (1936).
[9] Gilbert E. N. :, Gray codes and Paths on the n-cube, Bell System Tech. J., 37, 815–826 (1957).
[10] Gotsman C., Lindenbaum M.:, On the metric properties of discrete space-filling curves, IEEE Trans. on Image Processing, 5(5), 794–797 (1996).
[11] He Z. and Owen A. B.: Extensible grids: uniform sampling on a space filling curve, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 78(4), 917–935, (2016).
[12] Hilbert D. : Ueber die stetige Abbildung einer Linie auf ein Flaechenschtueck, Mathematische Annalen 38, 459–469, (1891).

[13] Kim, H.I.; Hong, S.; Chang, J.W.: Hilbert curve-based cryptographic transformation scheme for spatial query processing on outsourced private data. Data & Knowledge Engineering 104 (C), 32–44 (2016).
[14] Kumar G.S., Pandithevan P., Ambatti A.R.: Fractal raster tool paths for layered manufacturing of porous objects, Virtual and Physical Prototyping 4(2),91–104, (2009).
[15] Lawder J. K. and King P. J. H.:Using space-filling curves for multidimensional indexing, in Proc. 17th BNCOD, in Lecture Notes in Computer Science, 1832. 20–35, (2000)
[16] Lian, H., Qiu, W., Yan, D., Guo J., Li Z., Tang, P.: Privacy-preserving spatial query protocol based on the Moore curve for location-based service, Computers & Security, 96, 101845, (2020).
[17] Melian, S., Brix, K., Müller, S., Schieffer, G. (2011). Space-Filling Curve Techniques for Parallel, Multiscale-Based Grid Adaptation: Concepts and Applications. In: Kuzmin, A. (eds) Computational Fluid Dynamics 2010. Springer, Berlin, Heidelberg
[18] Milne S.C.:, Peano Curves and Smoothness of Functions, Advances in Mathematics 35 ,129-157 (1980).
[19] Moon, B., Jagadish, H.V., Faloutsos, C., Saltz, J.H.: Analysis of the clustering properties of the hilbert space-filling curve. IEEE Trans Knowl Data Eng 13 (1), 124–141 (2001). doi:10.1109/69.908985
[20] Moore E. H.:, On certain crinkly curves, Trans. Amer. Math. Soc., 1 (1900), pp. 72–90.
[21] Nair S.H., Sinha A., Vachhani L.: Hilbert's space-filling curve for regions with holes, Proc. of IEEE Conference on Decision and Control, 313–319 (2017).
[22] Ouni, T., Lassoued, A. and Abid M.: Lossless image compression using gradient based space filling curves (G-SFC). Signal, Image and Video Processing 9, 277–293 (2015).
[23] O'Shaughnessy S., Sheridan S.:Image-based malware classification hybrid framework based on space-filling curves, Computers & Security, 116, 102660, (2022).
[24] Patrick E.D., Anderson D.R., Bechtel F.K., Mapping multidimensional space to one dimension for computer output display, IEEE Trans. on Comput. 17, 949–953, (1968).
[25] Peano G.:, Sur une courbe qui remplit toute une aire plane, Math. Ann., 36, 157–160 (1890).
[26] L. K. Platzman and J. J. Bartholdi:, Spacefilling Curves and the Planar Traveling Salesman Problem, Journal of ACM, 36, 719–737 (1989).
[27] Sagan H., Space-filling Curves, Springer-Verlag, New York, 1994.
[28] Schrack G., Stocco L. : Generation of Spatial Orders and Space-Filling Curves, IEEE Trans. on Image Processing, 24(6, 1791–1800, (2015).
[29] S. Sharma, K. Hari and G. Leus, Space filling curves for MRI sampling in IEEE ICASSP, 1115–1119, (2020).
[30] Sierpiński W.:, O pewnej krzywej wypełniającej kwadrat. Sur une nouvelle courbe continue qui remplit toute une aire plane., Bulletin de l'Acad. des Sciences de Cracovie A.,463–478, (1912).
[31] Sergeyev, Y.D.: An information global optimization algorithm with local tuning. SIAM J. Optim. 5, 858–870 (1995)
[32] Sergeyev Y.D., Strongin R.G. , Lera D.: Introduction to Global Optimization Exploiting Space-Filling Curves, Springer New York Heidelberg Dordrecht London (2012).
[33] Sergeyev Y.D., ·Maria Chiara Nasso M.C.,Lera D.: Numericalmethods using two different approximations of space-filling curves for black-box global optimization, Journal of Global Optymization, published online 08.08.2022 https://doi.org/10.1007/s10898-022-01216-1, published online 08.08.2022.
[34] Skubalska-Rafajłowicz E.:, Applications of the space-filling curves with data driven measure–preserving property, Nonlinear Analysis, Theory, Methods and Applications, 30(3), 1305–1310, (1997).
[35] Skubalska-Rafajłowicz E., Rafajłowicz E.: Searching for optimal experimental designs using space-filling curves Applied Mathematics and Computer Science. 8(3) 647–656 (1998).
[36] Skubalska-Rafajłowicz E., Rafajłowicz E.: Space-filling curves in generating equidistrubuted sequences and their properties in sampling of images W: Signal processing / ed. by Sebastian Miron. Vukovar : In-Teh, 131–149 (2010.
[37] Skubalska-Rafajłowicz E.:, Krzywe wypełniające w rozwiązywaniu wielowymiarowych problemów decyzyjnych, Wydawnictwo Politechniki Wrocławskiej, 2001.
[38] Skubalska-Rafajłowicz E.:, Pattern recognition algorithm based on space-filling curves and orthogonal expansion, IEEE Trans. on Information Theory 47(5) 1915–1927, (2001).
[39] Skubalska-Rafajłowicz E.: A new method of estimation of the box-counting dimension of multivariate objects using space-filling curves. Nonlinear Analysis, Theory, Methods & Applications. Series A, Theory and Methods. 63 (5–7), 1281–1287,(2005).
[40] Skubalska-Rafajłowicz E., Rafajłowicz E.: Sampling multidimensional signals by a new class of quasi-random sequences. Multidimensional Systems and Signal Processing. 23 (1/2) 237–253, (2012).
[41] Steele J. M.: Efficacy of Spacefilling Heuristics in Euclidean Combinatorial Optimization, Operations Research Letters 8 , 237–239, (1989).
[42] Vogiatzis D. and Tsapatsoulis N.:Clustering Microarray Data with Space Filling Curves, in Proceedings of the 7th international workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory, LNAI 4578, 529–536, (2007).
[43] Zhou L., C. R. Johnson and D. Weiskopf: Data-Driven Space-Filling Curves, IEEE Transactions on Visualization and Computer Graphics, 27(2) 1591–1600, (2021).