Distrib Parallel Databases (2011) 29: 217–238
DOI 10.1007/s10619-010-7070-7

Irregularity in high-dimensional space-filling curves

Mohamed F. Mokbel · Walid G. Aref

Published online: 17 November 2010

Abstract A space-filling curve is a way of mapping the discrete multi-dimensional space into the one-dimensional space. It acts like a thread that passes through every cell element (or pixel) in the discrete multi-dimensional space so that every cell is visited exactly once. Thus, a space-filling curve imposes a linear order of the cells in the multi-dimensional space. There are numerous kinds of space-filling curves. The difference between such curves is in their way of mapping to the one-dimensional space. Selecting the appropriate curve for any application requires knowledge of the mapping scheme provided by each space-filling curve. *Irregularity* is proposed as a quantitative measure for the ordering quality imposed by space-filling curve mapping. The lower the irregularity the better the space-filling curve in preserving the order of the discrete multi-dimensional space. Five space-filling curves (the Sweep, Scan, Peano, Gray, and Hilbert) are analyzed with respect to irregularity. Closed formulas are developed to compute the irregularity in any dimension k for a D-dimensional space-filling curve with grid size N. A comparative study of different space-filling curves with respect to the irregularity is conducted and results are presented and discussed. We find out that for an application that is biased toward one of the dimensions, the Sweep or the Scan space-filling curves are the best choice. For high-dimensional applications, the Peano space-filling curve would be the best choice. For applications that require fairness among various dimensions, the Hilbert and Gray space-filling curves are the best choice.

Communicated by Hosagrahar Jagadish.

M.F. Mokbel (✉)
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
e-mail: mokbel@cs.umn.edu

W.G. Aref
Department of Computer Science, Purdue University, West Lafayette, IN, USA
e-mail: aref@cs.purdue.edu

Keywords Space-filling curves · Fractals · Irregularity · High-dimensional space · Performance analysis

1 Introduction

Mapping from the multi-dimensional space into the one-dimensional domain provides a pre-processing step for multi-dimensional applications. Examples of these applications include Multimedia databases, geographic information systems (GIS), QoS routing, and image processing. The main idea is to keep the existing algorithms and data structures independent of the data dimensionality. The objective of the mapping is to represent a point from the multi-dimensional space by a single integer value that reflects the various dimensions of the original space.

Space-filling curves (SFCs) have been extensively used as a mapping scheme from the *discrete* multi-dimensional space into the one-dimensional space (e.g., see [6, 9, 11, 15, 18, 24, 28, 30, 36, 44]). A space-filling curve is a thread that goes through all the points in the *discrete* space while visiting each point only one time. Thus, a space-filling curve imposes a linear order of points in the *discrete* multi-dimensional space. There are two different objectives of the mapping imposed by space filling curves, namely, *locality-preserving* mapping and *order-preserving* mapping. In *locality-preserving* mapping, if two points are near to each other in the multi-dimensional space, then they will be near to each other in the one-dimensional space. On the other side, in the *order-preserving* mapping, if two points are in a certain order in the multi-dimensional space with respect to a certain dimension, then they will be mapped in the same order in the one-dimensional space. The choice of a space-filling curve is mainly based on the underlying application. For example, locality-preserving mappings are more suitable for clustered-based applications (e.g., range queries, nearest-neighbor queries, declustering, and clustering). Order-preserving mappings are more suitable for sorting-based application (e.g., scheduling, indexing, and sorting).

In particular, declustering is an important issue in distributed and parallel databases. Good declustering can significantly improve the performance of database search queries when relations are distributed over several nodes, disks, or multiprocessors. Faloutsos and Bhagwat [16] have proposed to use the Hilbert space-filling curve to impose a linear order on a file of M units. Then, the units are traversed in their assigned order where each unit is assigned to a disk in a round-robin fashion.

In this paper, we go through three main steps:

1. We introduce the notion of *irregularity* as a quantitative measure of goodness for the order imposed by space-filling curves.
2. We analyze the behavior of five commonly used space-filling curves, namely, the Sweep, Scan, Peano, Gray, and Hilbert SFCs in the multi-dimensional space. These curves are given in Figs. 1 and 2 for the two-dimensional and three-dimensional spaces, respectively.
3. We develop closed formulas to compute the irregularity vector $V_I = (I_0, I_1, \ldots, I_{D-1})$, where I_k is the number of irregularities in dimension k, $0 \leq k \leq D-1$, and total irregularity $I_T(N, D) = \sum_{k=0}^{D-1} I_k$ for each space-filling curve with grid size N.

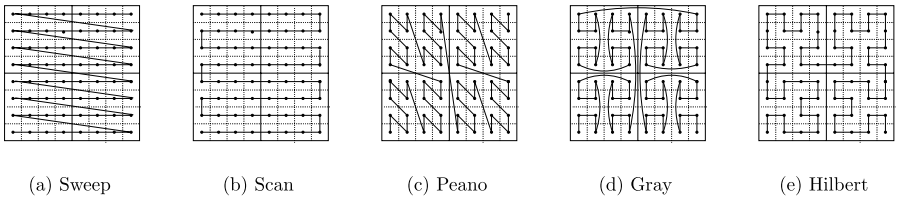


Fig. 1 Two-dimensional space-filling curves

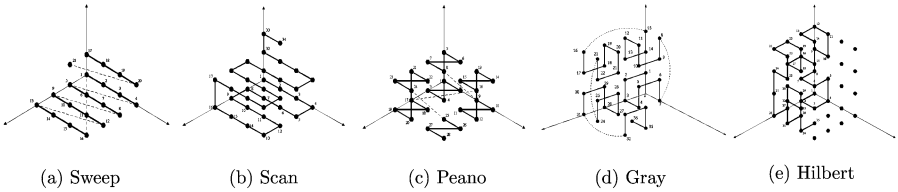


Fig. 2 Three-dimensional space-filling curves

The rest of this paper is organized as follows. Section 2 surveys related work. Section 3 introduces *irregularity* as a quantitative measure of goodness for the order imposed by space-filling curves. Section 4 analyzes the Sweep, Scan, Peano, Gray, and Hilbert SFCs, and develops closed formulas to compute the irregularity vector of each space-filling curve. Section 5 conducts a comprehensive comparison among different space-filling curves with respect to the irregularity. Finally, Sect. 6 concludes the paper.

2 Related work

Although space-filling curves were discovered in the nineteenth century [23, 38, 43], their use in computer science applications is not discovered until recently. The use of space-filling curves is motivated by the emergence of multi-dimensional applications. Space-filling curves are used for multi-dimensional spatial join [40]. Multi-dimensional data is transformed into the one-dimensional domain using the Z-order [41]. The transformed data is stored in a one-dimensional data structure, the B^+-Tree [14], and a one-dimensional spatial join algorithm [40] is applied. Multi-dimensional range queries utilize the clustering properties of the Gray [15] and Hilbert [24] SFCs to minimize the number of retrieved disk pages for the query answer. Space-filling curves are used as a spatial access method where the multi-dimensional data is stored in one-dimensional media (disk) using the Hilbert SFC [17, 18]. R-tree packing [27, 28] use the Hilbert SFC where a set of rectangles are sorted according to the Hilbert order, and then are packed into the R-Tree nodes.

The Z-order [41] is used as a spatial access method to enhance the performance of spatial join [11]. Spatial objects located in a disk are ordered according to their Z-order value to minimize the number of times a given page is retrieved from the disk. Similar use of space-filling curves is presented in the context of the

Hilbert SFC [44]. Other uses of space-filling curves include nearest-neighbor queries [30, 45], multi-dimensional indexing [29], indexing moving objects [12, 25], location privacy [20, 26], disk scheduling [5, 36], declustering [16], image processing [48, 50], memory management [46], the travelling salesman problem [7], bandwidth reduction for sampling signals [8], and graphics display generation [42].

Due to its wide applicability, numerous algorithms are developed for efficiently generating various space-filling curves. Recursive algorithms are proposed for generating the Hilbert SFC [10, 13, 21, 51] and the Peano SFC [13, 51]. Table-driven algorithm are proposed for the Peano and Hilbert SFCs [21]. The properties of different space-filling curves are studied extensively in the literature, e.g., see [1, 3, 4, 31, 33, 35, 37]. The clustering properties of the Hilbert SFC is analyzed by deriving closed formulas for the number of clusters in a given query region [37]. General functions that analyze the behavior of any space-filling curve in the multi-dimensional space is recently introduced [33, 35]. However, these functions and analysis are applied for medium dimensionality and do not reflect the quality of the sorting order imposed by an SFC.

The Spectral mapping is presented [34] as an alternative to fractal space-filling curves for locality-preserving mappings. The optimality of the spectral mapping is proved [32] in the sense it aims to minimize the one-dimensional distance between any two neighbor points in the multi-dimensional space. However, spectral mapping is not order-preserving where there is no guarantee that any two points in the multi-dimensional space would keep their order in the one-dimensional space.

3 Irregularity in space-filling curves

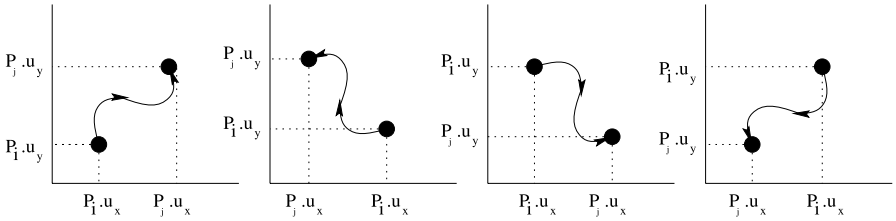
An optimal order-preserving space-filling curve is one that sorts multi-dimensional points in ascending order for all dimensions. However, in reality, when a space-filling curve attempts to sort the points in ascending order according to one dimension, it fails to do the same for the other dimensions. A good space-filling curve for one dimension is not necessarily good for the other dimensions. In order to measure the mapping quality of a space-filling curve, we introduce the concept of *irregularity* as a measure of goodness for the order imposed by a space-filling curve. Irregularity introduces a quantitative measure that indicates the non-avoidable reverse order imposed by space-filling curves for some or all dimensions. Irregularity is measured for each dimension separately, and gives an indicator of how a space-filling curve is far from the optimal. The lower the irregularity, the better the space-filling curve. Table 1 summarizes the used symbols in this paper.

Definition 1 For any two points, say P_i and P_j, in the D-dimensional space with coordinates $(P_i.u_0, P_i.u_1, \ldots, P_i.u_{D-1})$, $(P_j.u_0, P_j.u_1, \ldots, P_j.u_{D-1})$, respectively, and for a given space-filling curve S, if S visits P_i before P_j, we say that an irregularity occurs between P_i and P_j in dimension k iff $P_j.u_k < P_i.u_k$.

Figure 3 demonstrates all possible scenarios that can lead to an irregularity in the two-dimensional space, where the arrows in the curves indicate the order imposed by the underlying space-filling curve, i.e., point P_i is visited before point P_j.

Table 1 Symbols

Symbol	Description
D	Number of dimensions
N	Grid size; number of points in each dimensions
$I(k, N, D)$	Number of irregularities for dimension k in a D-dimensional with grid size N
$I_T(N, D)$	Total number of irregularities in all the D dimensions with grid size N
V_I	Irregularity vector



(a) No Irregularity in x, y (b) Irregularity in x only (c) Irregularity in y only (d) Irregularity in x, y

Fig. 3 Irregularity in two-dimensional space

In Fig. 3a $P_i.u_x < P_j.u_x$ and $P_i.u_y < P_j.u_y$, thus there is no irregularity in any of the dimensions. In Fig. 3b, $P_i.u_x > P_j.u_x$ which leads to only one irregularity in dimension x. Similarly, there is only one irregularity in dimension y in Fig. 3c where $P_i.u_y > P_j.u_y$. The worst-case scenario occurs in Fig. 3d where there are two irregularities in both dimensions.

Definition 2 For a given D-dimensional space-filling curve with grid size N, the number of irregularities for any dimension k is: $I(k, N, D) = \sum_{j=1}^{N^D} \sum_{i=1}^{j-1} f_{ij}$, where $f_{ij} = 1$ if $P_i.u_k > P_j.u_k$, and $f_{ij} = 0$, otherwise.

An optimal space-filling curve for any dimension k would have no irregularity, i.e., $I_{\text{optimal}}(k, N, D) = 0$. In contrast, the worst-case scenario for any dimension k is to sort all points in reverse order. The following lemma computes the number of irregularities in a worst-case scenario of any D-dimensional space-filling curve.

Lemma 1 *The worst-case scenario in dimension k of any D-dimensional space-filling curve with grid size N would have $I_{\text{worst}}(k, N, D) = \frac{1}{2}N^{2D-1}(N-1)$ irregularities.*

Proof The worst-case scenario for a dimension k takes place when all multi-dimensional points are projected on k and scanned in a reverse order. So, for any two values i, j where $0 \leq j < i \leq k-1$, j would result in $N^{D-1}N^{D-1}$ irregularities with i. Recall that there are N^{D-1} points of each value along the grid (a total of N^D points in the space). Since there is $i-1$ values less than i, then i would have

$N^{2D-2}i$ irregularity. Thus, the worst case irregularity would be $I_{\text{worst}}(k, N, D) = \sum_{i=0}^{N-1} N^{2D-2}i$. This can be simplified to: $I_{\text{worst}}(k, N, D) = \frac{1}{2}N^{2D-1}(N-1)$. □

4 Case studies

In this section, we derive closed formulas that compute the irregularity for any dimension k in a D-dimensional space with grid size N. We focus on five commonly used space-filling curves: the Sweep, Scan, Peano, Gray, and Hilbert SFCs. For each space-filling curve, we derive two formulas; the first formula gives the number of irregularities I_k in each dimension k. Hence, the irregularity vector $V_I = (I_0, I_1, \ldots, I_{D-1})$ can be computed. The second formula gives the total number of irregularities $I_T(N, D)$ over all dimensions.

4.1 Case study I: the Sweep SFC

Figures 1a and 2a give the Sweep SFC in the two- and three-dimensional spaces with grid sizes eight and four, respectively. Applications of the Sweep SFC include storing multi-dimensional arrays in memory and scheduling disk requests. A point in the D-dimensional Sweep SFC with grid size N can be represented by a D-digit number in a base-N system. The leftmost digit represents the last dimension ($k = D-1$), while the rightmost digit represents the first dimension ($k = 0$). Then, multi-dimensional points are visited in the order imposed by the base-N system. Thus, the first dimension is always keep changing from 0 to $N-1$. We define the *round* of any dimension in the Sweep SFC as the set of consecutive points that start with value 0 and end with value $N-1$. The last dimension has only one *round* with N^D points while the first dimension has N^{D-1} *rounds* with N points in each *round*.

Lemma 2 *The irregularity vector V_I for the D-dimensional Sweep SFC with grid size N is $V_I = (I_0, I_1, \ldots, I_{D-1})$ where*: $I_k = \frac{N^{D+k}}{4}(N^{D-k-1} - 1)(N-1)$.

Proof Considering the first dimension ($k = 0$), a point with value i in the jth ($j > 0$) *round* would have $(j-1)(N-i-1)$ irregularity. The term $(N-i-1)$ corresponds to the number of points in each round that have a value larger than i while the term $(j-1)$ represents the number of previous *rounds*. Given that the first dimension has N^{D-1} rounds, then $I(0, N, D) = \sum_{j=1}^{N^{D-1}} \sum_{i=0}^{N-1} (j-1)(N-i-1)$. This can be simplified to: $I(0, N, D) = \frac{N^D}{4}(N^{D-1} - 1)(N-1)$.

For higher dimensions ($k > 0$), the kth dimension in a D-dimensional space can be derived from the $(k-1)$th dimension in the $(D-1)$-dimensional space using the recursive relation: $I(k, N, D) = N^2 I(k-1, N, D-1)$. Solving this recursive relation, we get: $I_k = \frac{N^{D+k}}{4}(N^{D-k-1} - 1)(N-1)$. □

Lemma 3 *In a D-dimensional space with grid size N, the total number of irregularities over all dimensions for the Sweep SFC is*: $I_T(N, D) = \frac{N^{2D}}{4}(D-1) - \frac{N^D}{4}(DN^{D-1} - 1)$.

Proof Using Lemma 2, we have $I_T(N, D) = \sum_{k=0}^{D-1} I_k$. □

4.2 Case study II: the Scan SFC

The Scan SFC (Figs. 1b and 2b) is a slight variation of the original Sweep SFC. Similar to the Sweep SFC, the first dimension in the Scan SFC has N^{D-1} *rounds*, each with N points. In contrast to the Sweep SFC, the Scan SFC distinguishes between even- and odd-numbered *rounds*. *Rounds* are numbered from 0 to $N^{D-1} - 1$. Even-numbered *rounds* are the same as those of the Sweep SFC where points are visited in the order 0 to $N - 1$. However, the odd-numbered *rounds* are visited in the reverse order from $N - 1$ to 0.

Lemma 4 *The irregularity vector V_I for the D-dimensional Scan SFC with grid size N is $V_I = (I_0, I_1, \ldots, I_{D-1})$ where: $I_k = \frac{N^{2D-1}}{4}(N-1)$ for $k < D-1$, and $I_{D-1} = 0$.*

Proof Since the Scan SFC has the same concept of *round* as that of the Sweep SFC, the irregularity in the Sweep SFC is inherited into the Scan SFC. In addition, odd-numbered *rounds* in the Scan SFC results in an additional irregularity $I_{\text{odd}}(k, N, D)$ due to their reversed behavior. Using the result of Lemma 2, the irregularity in the first dimension $k = 0$ for the Scan SFC is: $I(0, N, D) = \frac{N^D}{4}(N^{D-1} - 1)(N - 1) + I_{\text{odd}}(0, N, D)$.

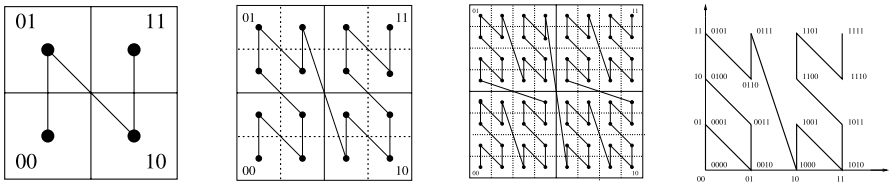
A point with value i in an odd *round* has $(N - i - 1)$ irregularity within the *round*. Thus, the number of irregularities inside each odd *round* is $\sum_{i=0}^{N-1}(N - i - 1) = \frac{N(N-1)}{2}$. Given that half of the *rounds* are odd-numbered, then $I_{\text{odd}}(0, N, D) = \frac{N^D(N-1)}{4}$. Therefore: $I(0, N, D) = \frac{N^{2D-1}}{4}(N - 1)$. Similar to the Sweep SFC, the last dimension in the Scan SFC has no irregularity: $I(D - 1, N, D) = 0$. Other dimensions follow the recursive relation: $I(k, N, D) = N^2 I(k - 1, N, D - 1)$ for $k < D - 1$. Solving this recurrence relation: $I_k = \frac{N^{2D-1}}{4}(N - 1)$ for $k < D - 1$, and $I_{D-1} = 0$. □

Lemma 5 *In a D-dimensional space with grid size N, the total number of irregularities over all dimensions for the Scan SFC is: $I_T(k, N, D) = \frac{N^{2D-1}}{4}(N-1)(D-1)$.*

Proof Using Lemma 4, we have $I_T(N, D) = \sum_{k=0}^{D-1} I_k$. □

4.3 Case study III: the Peano SFC

The Peano SFC (Figs. 1c and 2c) is introduced by Peano [43] and is also termed Morton encoding [39], quad code [19], bit-interleaving [47], N-order [49], locational code [2], or Z-order [41]. The Peano SFC is constructed recursively as in Fig. 4. The basic shape (Fig. 4a) contains four points in the four quadrants of the space. Each quadrant is represented by two binary digits. The most significant digit is represented by its x position while the least significant digit is represented by its y position. The Peano SFC orders space quadrants in ascending order (00, 01, 10, 11). Figure 4b



(a) N=2 (b) N=4 (c) N=8 (d) Bit-interleaving

Fig. 4 The Peano SFC

Table 2 An example of two- and three-dimensional Peano orders with grid size 8 in each dimension

Point	Dimensions		Bit-interleaving	Decimal Order	Point	Dimensions			Bit-interleaving	Decimal Order
	0	1				0	1	2		
(2, 1)	**010**	001	**001001**	9	(0, 1, 3)	**000**	*001*	011	**0000010***11*	11
(5, 3)	**101**	011	**100111**	39	(2, 1, 4)	**010**	*001*	100	**001100***10*	98
(7, 0)	**111**	000	**101010**	42	(7, 0, 7)	**111**	*000*	111	**101101***01*	365

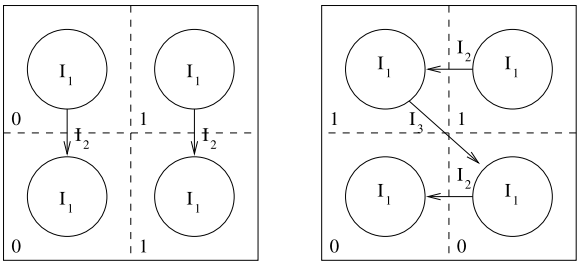
contains four blocks of Fig. 4a at a finer resolution and is visited in the same order as in Fig. 4a. Similarly, Fig. 4c contains four blocks of Fig. 4b at a finer resolution.

Figure 4d represents the bit interleaving in the two-dimensional Peano SFC. Each point is represented by a binary number that results from interleaving bits of the two dimensions. The bits are interleaved according to an interleaving vector $T_v = (0, 1, 0, 1)$. T_v indicates that the first and third bits are taken from dimension 0 (x) while the second and fourth bits are taken from dimension 1 (y). For a D-dimensional space with four points in each dimension (i.e., $N = 4$), the interleaving vector is $T_v = (0, 1, 2, \ldots, D-1, 0, 1, 2, \ldots, D-1)$. For a grid size of N points in each dimension, the term $0, 1, 2, \ldots, D-1$ is repeated $\mathrm{Log}\, N$ times. The points are visited in ascending order according to their binary number representation. Table 2 gives an example of computing the Peano order for two- and three-dimensional points with a grid size of eight points in each dimension.

Lemma 6 *The irregularity vector V_I for the D-dimensional Peano SFC with grid size N is $V_I = (I_0, I_1, \ldots, I_{D-1})$ where: $I_k = \frac{N^D(N^D-1)(2^D-2^{D-k-1}-1)}{4(2^D-1)} - \frac{N^D(N^{D-1}-1)}{4}$.*

Proof Considering the two-dimensional Peano SFC in Fig. 4, the four blocks are visited in the linear order $L_P = (00, 01, 10, 11)$. We define a vector v_k for dimension k as the projection of L_P on k. So, $v_0 = (0, 0, 1, 1)$ and $v_1 = (0, 1, 0, 1)$. Then, we can distinguish the following three sources of irregularities (Fig. 5):

1. $I_{P1}(k, N, D)$: The irregularity of the Peano SFC in the D-dimensional space within each subblock of grid size $N/2$ (I_1 in Fig. 5).
2. $I_{P2}(k, N, D)$: For each v_k, the blocks with value zero/one affect other blocks with the same value (I_2 in Fig. 5).



(a) Block Irregularity for x (b) Block Irregularity for y

Fig. 5 Block irregularity for x, y dimensions in the two-dimensional Peano SFC

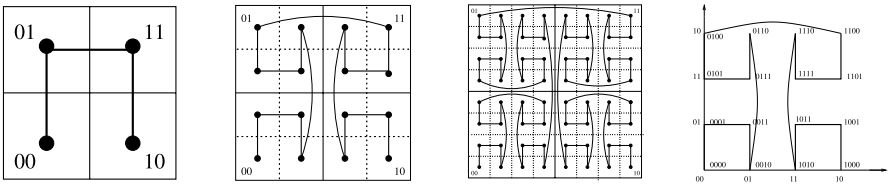
3. $I_{P3}(k, N, D)$: For each v_k, the blocks with value one affects blocks with value zero. In Fig. 5a, I_3 does not appear in dimension x since $v_x = (0, 0, 1, 1)$. For dimension y, $v_y = (0, 1, 0, 1)$, there is a block with value one that affects a block with value zero (I_3 in Fig. 5b).

The total irregularity $I_P(k, N, D)$ is the sum of the previous three irregularities: $I_P(k, N, D) = I_{P1}(k, N, D) + I_{P2}(k, N, D) + I_{P3}(k, N, D)$.

$I_{P1}(k, N, D)$: For a D-dimensional space with grid size N, there are 2^D blocks with grid size $N/2$. Therefore, $I_{P1}(k, N, D) = 2^D I_P(k, N/2, D)$.

$I_{P2}(k, N, D)$: In a D-dimensional space with grid size N, each block contains $(N/2)^D$ points distributed evenly over $N/2$ values, i.e., there are $(N/2)^{D-1}$ points for each value. Consider any two blocks b_1 and b_2 with the same value (0/1) in any v_k, and b_1 is visited before b_2. A point with value i in b_2 would have $(N/2)^{D-1} * (N/2)^{D-1} * (N/2 - i - 1)$ irregularities. The first $(N/2)^{D-1}$ represents the number of points with value i in b_2. The second $(N/2)^{D-1}$ represents the number of points with value j in b_1, where $i < j$. The term $(N/2 - i)$ represents the j values where $i < j$. Summing over all the values, the irregularity between any two blocks is: $\sum_{i=0}^{N/2-1} (N/2)^{D-1} * (N/2)^{D-1} * (N/2 - i - 1)$. Since each dimension contains 2^D blocks half of them with value one, and the other half with value zero, therefore, we have 2^{D-1} blocks with value zero. The jth visited block with value zero is affected by the next $2^{D-1} - j - 1$ blocks. There is an equal number of blocks with value one as with value zero, so the total irregularity is multiplied by two. Thus, $I_{P2}(k, N, D) = 2(\frac{N}{2})^{D-1}(\frac{N}{2})^{D-1} \sum_{i=0}^{N/2-1} (N/2 - i - 1) \sum_{j=0}^{2^{D-1}-1} (2^{D-1} - j - 1)$. This can be simplified to: $I_{P2}(k, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1)$.

$I_{P3}(k, N, D)$: The last part of the irregularity is the effect of visiting blocks with value one in v_k before blocks with value zero in v_k. As a result, all points in the *zero* block will have irregularity with all points in the *one* block. So, the irregularity between any two such blocks is $(N/2)^D * (N/2)^D$. Any vector v_k has a base sequence s_k that is the maximum sequence of consecutive *zero* blocks followed by consecutive *one* blocks. v_k is constructed by concatenating several copies of s_k. For example, in the three-dimensional Peano SFC, the third dimension ($k = 2$) has $s_2 = (01)$, v_2 is constructed by four copies of s_2, i.e., $v_2 = (01010101)$. Similarly, $s_1 = (0011)$ and $s_0 = (00001111)$. In the D-dimensional space, for any dimension k, s_k contains 2^{D-k-1} zeros followed by 2^{D-k-1} ones, v_k is constructed by hav-



(a) N=2 (b) N=4 (c) N=8 (d) Bit-interleaving

Fig. 6 The Gray SFC

ing 2^k copies of s_k. The ith copy of s_k results in $(N/2)^D * (N/2)^D * 2^{D-k-1} * 2^{D-k-1} * (2^k - i - 1)$ irregularities. Therefore, $I_{P3}(k, N, D) = (\frac{N}{2})^D (\frac{N}{2})^D (2^{D-k-1} * 2^{D-k-1}) \sum_{i=0}^{2^k-1} 2^k - i - 1$. That can be simplified to: $I_{P3}(k, N, D) = \frac{2^k-1}{2^{k+3}} N^{2D}$.

By combining the equations for I_{P1}, I_{P2}, and I_{P3} we obtain the following recurrence relation: $I_P(k, N, D) = 2^D I_P(k, \frac{N}{2}, D) + 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1) + \frac{2^k-1}{2^{k+3}} N^{2D}$, $I_P(k, 1, D) = 0$, where there is no irregularity when $N = 1$. Solving this recurrence relation results in: $I_k = \frac{N^D}{4(2^D-1)}((N^D - 1)(2^D - 2^{D-k-1} - 1) - (N^{D-1} - 1)(2^D - 1))$. □

Lemma 7 *In a D-dimensional space with grid size N, the total number of irregularities over all dimensions for the Peano SFC is*: $I_T(N, D) = \frac{N^D}{4}(DN^{D-1}(N - 1) - N^D + 1)$.

Proof Using Lemma 6, we have $I_T(N, D) = \sum_{k=0}^{D-1} I_k$. □

4.4 Case study IV: the Gray SFC

The Gray SFC (Figs. 1d and 2d) uses the Gray code representation [22] in contrast to the binary code representation as in the Peano SFC. Figure 6 gives the recursive construction of the Gray SFC. The basic shape (Fig. 6a) contains four points in the four quadrants of the space. The Gray SFC visits the space quadrants in ascending order according to the Gray code (00, 01, 11, 10). Figure 6b is constructed by having the first and fourth blocks as those of Fig. 6a, while the second and the third blocks are the rotation of the blocks in Fig. 6a by 180^0. Similarly, Fig. 6c is constructed from two blocks of Fig. 6b at a finer resolution and two blocks of the rotation of Fig. 6b by 180°.

Figure 6d represents the bit interleaving in the two-dimensional Gray SFC. Table 3 gives an example of computing the Gray order for two- and three-dimensional points with grid size eight (i.e., eight points) in each dimension.

Lemma 8 *The irregularity vector V_I for the D-dimensional Gray SFC with grid size N is $V_I = (I_0, I_1, \ldots, I_{D-1})$ where*: $I_0 = \frac{N^{2D-1}}{4}(\frac{N}{2} - 1)$, *and* $I_k = \frac{N^{2D-1}}{4}(N - 1)$ *for* $k > 0$.

Table 3 An example of two- and three-dimensional Gray orders with grid size 8 in each dimension

Point	Dimensions		Bit-interleaving	Decimal Order	Point	Dimensions			Bit-interleaving	Decimal Order
	0	1				0	1	2		
(2, 1)	**011**	001	**0**0**1**0**1**1	13	(0, 1, 3)	**000**	*001*	010	**0**0**0**0**1**0*1*0	12
(5, 3)	**111**	010	**1**0**1**1*1*0	52	(2, 1, 4)	**011**	*001*	110	**0**0*1***1**0*1*0	75
(7, 0)	**100**	000	**1**0**0**0**0**0	63	(7, 0, 7)	**100**	*000*	100	**1**0**0**0**0**1*0*0	384

Proof Considering the two-dimensional Gray SFC in Fig. 6, the four blocks are visited in the linear order $L_G = (00, 01, 11, 10)$. We define a vector v_k for dimension k as the projection of L_G on k. So, $v_0 = (0, 0, 1, 1)$ and $v_1 = (0, 1, 1, 0)$. Similar to the Peano SFC, the irregularity in the Gray SFC has three component: $I_G(k, N, D) = I_{G1}(k, N, D) + I_{G2}(k, N, D) + I_{G3}(k, N, D)$.

$I_{G1}(k, N, D)$: In the D-dimensional Gray SFC, the blocks of grid size N are composed of 2^{D-1} blocks b of grid size $N/2$ and 2^{D-1} blocks b_r as the rotation of b by 180^0 (Fig. 6). For any dimension k, the points in b_r are visited in the reverse order of those of b. For any pair of blocks (b, b_r) with grid size $N/2$, the total number of irregularities caused by these two blocks is the worst-case irregularity $I_{\text{worst}}(k, N/2, D)$ (Lemma 1). Since we have 2^{D-1} such pairs, therefore, $I_{G1}(k, N, D) = 2^{D-1} I_{\text{worst}}(k, N/2, D)$. That can be simplified to: $I_{G1}(k, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)$.

$I_{G2}(k, N, D)$: The second component of irregularity is the effect of blocks with similar value (zero or one) on each other. Since, we have the same number of zeros/ones as in the Peano SFC, then, we have: $I_{G2}(k, N, D) = I_{P2}(k, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1)$.

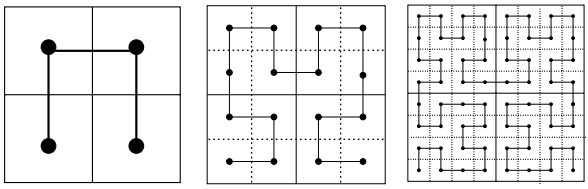
$I_{G3}(k, N, D)$: The third component of irregularity is the effect of the *one* blocks on the *zero* blocks. As a property of the Gray code [22], the number of one blocks that affect the zero blocks for all dimensions $k > 0$ is independent of k. For example, in the three-dimensional Gray SFC, $v_1 = (00111100)$, $v_2 = (01100110)$ results in eight irregular blocks. Based on this property, it is enough to get the irregularity $I_{G3}(k, N, D)$ for the second dimension only ($k = 1$) and the result will be valid for any $k > 0$. v_1 starts by 2^{D-2} zeros, followed by 2^{D-1} ones, then 2^{D-2} zeros. Each block contains $(N/2)^D$ points resulting in $(N/2)^D * (N/2)^D * 2^{D-1} * 2^{D-2}$ irregularities. Thus, $I_{G3}(0, N, D) = 0$, and $I_{G3}(k, N, D) = \frac{N^{2D}}{8}$ for $k > 0$.

Combining the equations for I_{G1}, I_{G2}, and I_{G3} results in: $I_G(0, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1) + 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1)$; $I_G(k, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1) + 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1) + \frac{N^{2D}}{8}, k > 0$. That can be simplified to: $I_0 = \frac{N^{2D-1}}{4}(\frac{N}{2} - 1)$, and $I_G = \frac{N^{2D-1}}{4}(N - 1), k > 0$. □

Lemma 9 *In a D-dimensional space with grid size N, the total number of irregularities over all dimensions for the Gray SFC is*: $I_T(N, D) = \frac{N^{2D-1}}{4}(DN - D - \frac{N}{2})$.

Proof Using Lemma 8, we have $I_T(N, D) = \sum_{k=0}^{D-1} I_k$. □

Fig. 7 The Hilbert SFC



(a) N=2 (b) N=4 (c) N=8

4.5 Case study V: the Hilbert SFC

Figure 7 gives the recursive construction of the Hilbert SFC. The basic block of the Hilbert SFC (Fig. 7a) is the same as that of the Gray SFC (Fig. 6a). The basic block is repeated four times at a finer resolution in the four quadrants, as given in Fig. 7b. The quadrants are visited in their gray order. The second and third blocks in Fig. 7b have the same orientation as in Fig. 7a. The first block is constructed from rotating the block of Fig. 7a by 90^0, while the fourth block is constructed by rotating the block of Fig. 7 by -90^0. Similarly, Fig. 7c is constructed from Fig. 7b.

Lemma 10 *The irregularity vector V_I for the D-dimensional Hilbert SFC with grid size N is $V_I = (I_0, I_1, \ldots, I_{D-1})$ where*: $I_0 = \frac{N(2^{2D-2}-1)}{4}(\frac{N^{2D-1}-1}{2^{2D-1}-1} - \frac{N^{2D-2}-1}{2^{2D-2}-1})$, *and* $I_k = \frac{N^{2D-1}}{4}(N-1)$ *for* $k > 0$.

Proof As in the Gray SFC, the Hilbert SFC has special treatment for the first dimension ($k = 0$) while all other dimensions are treated evenly. For all other dimensions $k > 0$, the proof of irregularity is the same as in the Gray SFC, so: $I(k, N, D) = \frac{N^{2D-1}}{4}(N-1)$ for $k > 0$.

The first dimension in the Hilbert SFC with grid size N is composed of 2^D blocks of the Hilbert SFC with grid size $N/2$. Only two of these blocks come from the first dimension at grid size $N/2$. So, the total number of irregularities can be written in the following recursive relation: $I_H(0, N, D) = 2I_H(0, N/2, D) + (2^D - 2)I(k, N/2, D) + I_{H2}(k, N, D)$, where $I_{H2}(k, N, D)$ is the irregularity that comes from arranging different blocks together. With the same proof as in the Peano and Gray SFCs: $I_{H2}(k, N, D) = I_{G2}(k, N, D) = I_{P2}(k, N, D) = 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1)$. So, we can write the following recurrence relation as: $I_H(0, N, D) = 2I_H(0, N/2, D) + (2^D - 2)I(k, N/2, D) + 2^{D-2}(\frac{N}{2})^{2D-1}(\frac{N}{2} - 1)(2^{D-1} - 1)$, $I(0, 2, D) = 0$. Solving the recurrence relation results in: $I_0 = \frac{N(2^{2D-2}-1)}{4}(\frac{N^{2D-1}-1}{2^{2D-1}-1} - \frac{N^{2D-2}-1}{2^{2D-2}-1})$. □

Lemma 11 *In a D-dimensional space with grid size N, the total number of irregularities over all dimensions for the Hilbert SFC is*: $I_T(N, D) = \frac{DN^{2D-1}(N-1)}{4} - \frac{2^{2D}N(N^{2D-1}-1)}{16(2^{2D-1}-1)}$.

Proof Using Lemma 10, we have $I_T(N, D) = \sum_{k=0}^{D-1} I_S(k, N, D)$. □

Table 4 The normalized irregularity and total irregularity for each space-filling curve

SFC	Percent of $I(k, N, D)$	Percent of $I_T(N, D)$
Sweep	$50(1 - N^{k-D+1})$	$50 - \frac{50(N^D-1)}{DN^{D-1}(N-1)}$
Scan	$I(k, N, D) = 50, k < D - 1$	$\frac{50(D-1)}{D}$
	$I(D-1, N, D) = 0$	
Peano	$50 - \frac{25\times 2^{D-k}(N^D-1)}{N^{D-1}(N-1)(2^D-1)}$	$50 - \frac{50(N^D-1)}{DN^{D-1}(N-1)}$
Gray	$I(0, N, D) = \frac{25(N-2)}{N-1}$	$50 - \frac{25N}{D(N-1)}$
	$I(k, N, D) = 50, k > 0$	
Hilbert	$I(0, N, D) = \frac{50(2^{2D-2}-1)}{N^{2D-2}(N-1)}(\frac{N^{2D-1}-1}{2^{2D-1}-1} - \frac{N^{2D-2}-1}{2^{2D-2}-1})$	$50 - \frac{25\times 2^{2D-1}(N^{2D-1}-1)}{DN^{2D-2}(N-1)(2^{2D-1}-1)}$
	$I(k, N, D) = 50, k > 0$	

5 Performance evaluation

In this section, we perform comprehensive experiments to compare the performance of the Sweep, Scan, Peano, Gray, and Hilbert SFCs with respect to irregularity. For comparison purposes, we compute the percentage of irregularity and percentage of total irregularity for each space-filling curve. The percentage of irregularity is computed by normalizing the number of irregularities in each space-filling curve (the closed formulas in Sect. 4) by the upper bound irregularity from Lemma 1. The percentage of total irregularity is computed as the average percentage of irregularity over all dimensions. Table 4 gives the closed formulas for the percentage of irregularity and percentage of total irregularity for the Sweep, Scan, Peano, Gray, and Hilbert SFCs. It is important to note that all the experiments in this section discuss the properties of each space-filling curve, regardless of the underlying data. At the end, space-filling curves map the multi-dimensional space into a one-dimensional space regardless of where the data is located in the multi-dimensional space. If the data distribution is uniform/skewed in the multi-dimensional space, the mapped data will sill be uniform/skewed in the one-dimensional space.

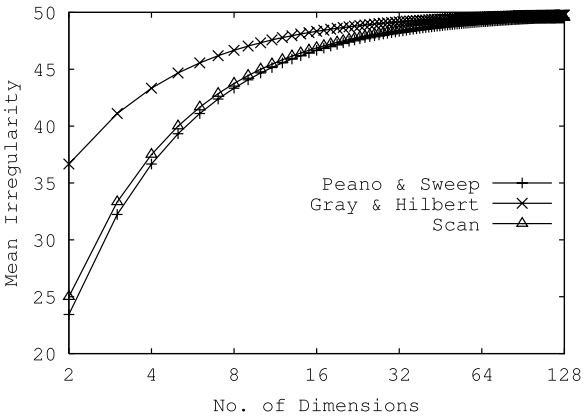
5.1 Scalability of space-filling curves

In this section, we address the issue of scalability, i.e., when the number of dimensions increases or when the number of points per dimension increases. The objective of the following experiments is to get the threshold values for dimensionality t_d and grid size t_g for each space-filling curve such that increasing the dimensionality and/or the grid size over t_d and t_g, respectively, does not significantly affect the percentage of irregularity for a space-filling curve. Table 5 gives an upper bound of the percentage of total irregularity $I_T(N, D)$ with the increase of dimensionality and grid size.

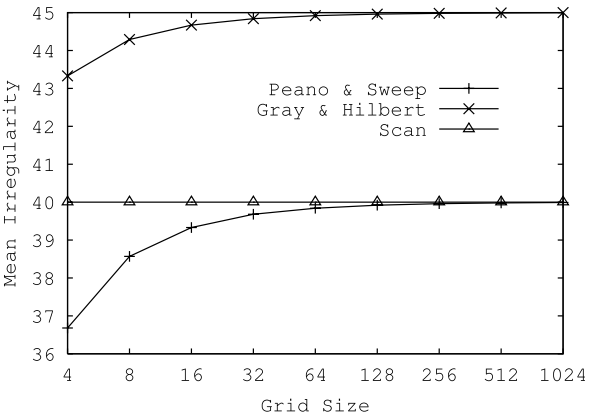
Figure 8a gives the percentage of irregularity for up to 128 dimensions with grid size 16 (the x axis is drawn in a log scale). According to Lemmas 3 and 7 the Sweep and Peano SFCs have exactly the same performance. Also from Lemmas 9 and 11, the Gray and Hilbert SFCs almost have the same performance. For low-dimensionality

Table 5 The upper bound of the percentage of total irregularity with respect to D and N

SFC	$\lim_{D\to\inf} I_T(N, D)$	$\lim_{N\to\inf} I_T(N,D)$
Sweep	50	$50 - \frac{50}{D}$
Scan	50	$\frac{50(D-1)}{D}$
Peano	50	$50 - \frac{50}{D}$
Gray	50	$50 - \frac{25}{D}$
Hilbert	50	$50 - \frac{25}{D}\frac{2^{2D-1}}{2^{2D-1}-1}$



(a) Dimensions



(b) Grid Size

Fig. 8 Scalability of space-filling curves

(up to 10 dimensions), the Peano and Sweep SFCs have the best performance. The performance of the Scan SFC approaches the Peano and Sweep SFCs for medium dimensionality (10 to 40 dimensions). In general the performance of the Gray and Hilbert SFCs is the worst for up to 64 dimensions. All space-filling curves tend to have the same performance for more than 64 dimensions. Notice that the percentage of irregularity never exceeds 50% for any space-filling curve. The 50% irregularity

matches with the analytical results in Table 5. From Fig. 8a, the threshold value t_d for all space-filing curves is 128.

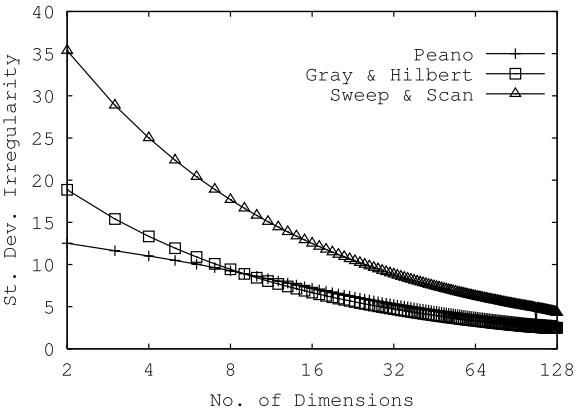
Figure 8b gives the result of the same experiment as that of Fig. 8a for the five-dimensional space while increasing the grid size for up to 1024. As indicated in Table 5, the steady state value for both the Gray and Hilbert SFCs is 45, while for the Sweep, Scan and Peano SFCs is 40. The performance of the Gray and Hilbert SFCs is worse than those of the Sweep, Scan and Peano SFCs for any grid size. For grid sizes less than 128, the Peano and Sweep SFCs have better performance than the Scan SFC. Notice that the performance of the Scan SFC is constant regardless of N, which is also reflected in Lemma 5 where the irregularity does not depend on N. The threshold value for grid size t_g is 128 for both the Gray and Hilbert SFCs, 4 for the Scan SFC, and 256 for both the Sweep and Peano SFCs.

5.2 Fairness of space-filling curves

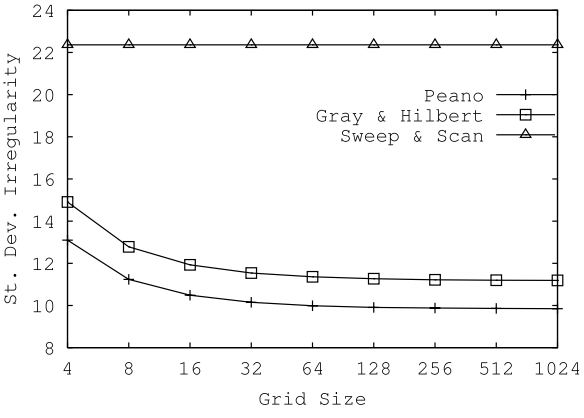
In this section, we test the fairness of space-filling curves. We say that a space-filling curve is fair if it has similar behavior towards all dimensions in the multi-dimensional space. We use the standard deviation of the percentage of irregularity over all dimensions as our measure of performance. The lower the standard deviation the more fair the space-filling curve is. For the experiments in this section, we use $I(k, N, D)$ from Table 4 to compute the percentage of irregularity for each dimension k. Figure 9a gives the fairness of each space-filling curve for up to 128 dimensions with grid size 16. The Sweep and Scan SFCs have almost the same performance, which is worse than all the other space-filling curves. The main reason is that both the Sweep and Scan SFCs have zero irregularity in the last dimension (Lemmas 3 and 5) while they have a high irregularity in the other dimensions. This variation of irregularity results in a high standard deviation. However, with the increase of dimensionality, the effect of the last dimension in the standard deviation is decreased. Thus, more fairness is produced with the dimensionality increase.

The Gray and Hilbert SFCs have the same irregularity performance where they have 50% irregularity for all dimensions except the first. For low dimensionality (up to 8 dimensions), the Gray and Hilbert SFCS have a performance that is worse than the Peano SFCs due to the first dimension that does not have 50% irregularity. With the increase in dimensionality, the effect of the first dimension in both the Gray and Hilbert SFCs is decreased, thus producing more fairness. For more than 10 dimensions, the Gray and Hilbert SFCs have the best fairness performance. As the dimensionality increases, the performance of the Gray and Hilbert SFCs is increased. As a steady state value, the standard deviation of the irregularity for the Gray and Hilbert SFC tends to be zero. The Peano SFC has the best performance for low dimensionality. Although the performance of the Peano SFC increases with the increase of dimensionality, it does not have the same incremental rate as those of the Gray and Hilbert SFCs.

Figure 9b performs the same experiment as in Fig. 9a for the five-dimensional space while increasing the grid size for up to 1024. The Peano SFC gives the best fairness for all grid sizes. The Sweep and Scan SFCs have the worst performance for all grid sizes. Regardless of the grid size, the Sweep and Scan SFCs have constant



(a) Dimensions



(b) Grid Size

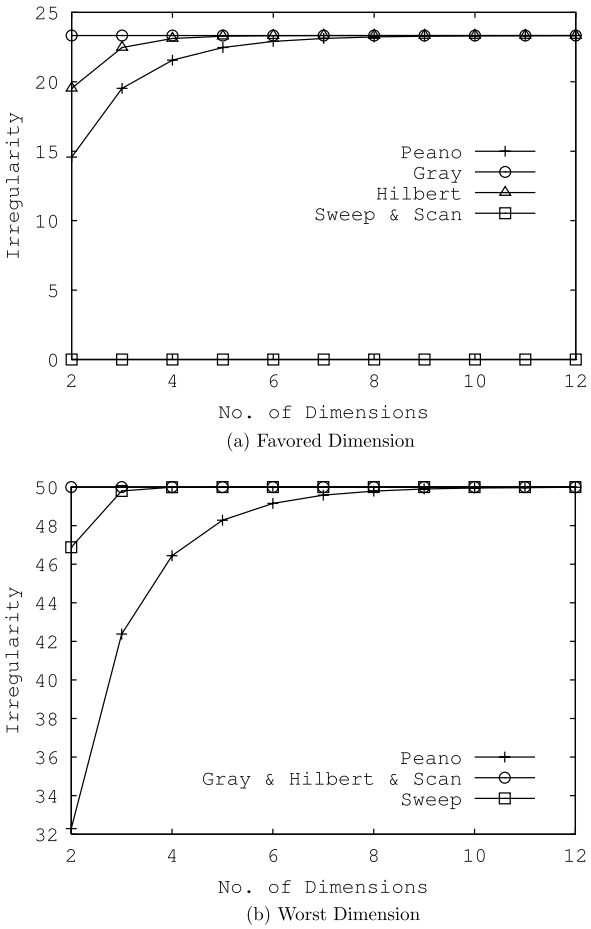
Fig. 9 Fairness of space-filling curves

behavior. The other space-filling curves tend to have a constant performance after grid size 128.

The experiments in this section are mainly useful for those applications that require fairness among different dimensions. For example, in a multimedia application where images and videos are represented as high-dimensional vectors, a fair space-filling curve is needed to layout these images or videos on the disk storage. As in Fig. 9a, the Gray and Hilbert SFCs will be most appropriate.

5.3 Intentional bias of space-filling curves

A critical point for SFC-based applications is how to assign the different parameters to the space dimensions. In this section, we explore the intentional bias of each space-filling curve towards one or more dimensions. We say that an SFC is intentionally biased towards a certain dimension k if the SFC has relatively higher irregularity in dimension k with respect to other dimensions. Some applications may have only one important dimension, while the other dimensions are not with the same signif-



(a) Favored Dimension

(b) Worst Dimension

Fig. 10 Favored and worst dimensions for space-filling curve irregularity

icant importance. For example, in real-time applications, the most significant issue is to satisfy the deadlines, other factors are not with the same importance. Also, in disk scheduling, the most important factor is to optimize the disk head movement. Assigning the appropriate dimension to the disk head has a great influence on the performance of disk scheduling [5]. For such applications, we develop the experiment given in Fig. 10. In Fig. 10a, the plotted dimension is the most favored dimension for each space-filling curve, e.g., the one with the lowest percentage of irregularity. In Fig. 10b, we plot the worst dimension, e.g., the one with the highest percentage of irregularity. Figure 10 gives the experiments for up to 12 dimensions with grid size 16. All space-filling curves tend to have a constant performance after 10 dimensions. Increasing the space dimensionality has no effect on performance. Thus, we consider the 12-dimensional space results as the steady state performance.

An interesting result is that the Scan and Sweep SFCs have constant percentage of irregularity 0% for the favored dimension (Fig. 10a) and 50% (Fig. 10b) for the worst dimension. This is the main reason of why both the Scan and Sweep SFCs have the worst fairness performance among all space-filling curves, as discussed in Sect. 5.2.

For the recursive space-filling curves, the Peano SFC has the best performance, where the percentage of irregularity ranges between 15% and 32% for low-dimensionality and between 24% and 50% for the steady state value. The Gray SFC has the worst performance where it has constant behavior for both the favored and worst dimensions, 24% and 50%, respectively.

The experiments in this section are mainly useful for those applications that tend to favor one particular dimension over all other dimensions. For example, in real-time applications, the time deadline has much higher priority than any other dimension. In this case, we may use either the Sweep or Scan SFC and assign the deadline dimension as the first dimension.

5.4 Irregularity in each single dimension

Figure 11 gives the percentage of irregularity for each dimension of the five-dimensional space with grid size 16 for the Peano, Gray, Hilbert, Sweep, and Scan SFCs. For the Peano SFC (Fig. 11a), there is a significant difference among all dimensions. This property makes the Peano SFC suitable for applications that have different parameters with different levels of priorities. The Peano SFC favors the dimensions on some ascending order. The relative difference of performance between any two consecutive dimensions decreases as the number of dimensions increases, e.g., the difference between the third and fourth dimensions is less than the difference between the first and second dimensions. The Gray SFC favors only one dimension (Fig. 11b), namely, the first dimension while dealing with the remaining dimensions fairly though with high irregularity. All dimensions except the first have 50% irregularity. The Hilbert SFC (Fig. 11c) has almost the same performance of the Gray SFC. Figure 11d gives the performance of the Sweep SFC. The first dimension has the best performance (0% irregularity), while the second dimension has irregularity 37% for grid size 4. The irregularity increases with the grid size till it reaches 50% for grid size 128. All other dimensions reaches 50% irregularity with grid size 16. The Scan SFC (Fig. 11e) has constant performance regardless of the grid size. Only the first dimensions has 0% irregularity, while all other dimensions have 50% irregularity.

This experiment is particularly useful when deciding about which dimensions will be assigned to which factors. For example, consider the problem of disk scheduling in multimedia servers [5, 36]. In addition to maximizing the bandwidth of the disk, the scheduler has to take into consideration the real-time constraints of the page requests, e.g., as in the case of video streaming. If clients are prioritized based on quality-of-service guarantees, then the disk scheduler might as well consider the priority of the requests in its disk queue. Scheduler parameters can be assigned different priorities and mapped to space dimensions based on the results in Fig. 11.

6 Conclusions

In this paper, we introduced the notion of *irregularity* as a quantitative measure of the ordering quality of space-filling curve mappings. Five space-filling curves (the Sweep, Scan, Peano, Gray, and Hilbert SFCs) are thoroughly analyzed with respect to

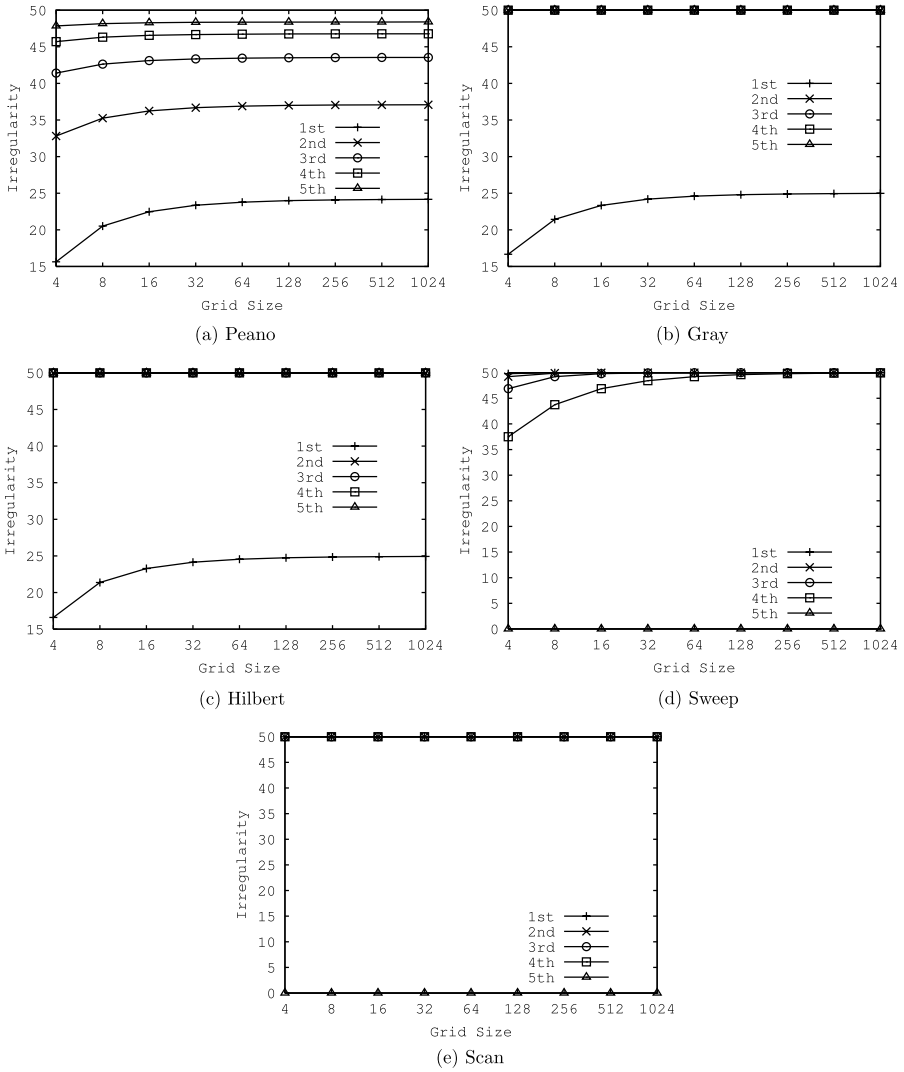


Fig. 11 Irregularity for all dimensions

irregularity. Each D-dimensional space-filling curve with grid size N is described by its irregularity vector $V_I = (I_0, I_1, \ldots, I_{D-1})$, where I_k is the number of irregularities in dimension k and its total irregularity $I_T(N, D)$. To avoid the brute force approach for computing the irregularity vector V_I and the total irregularity I_T, closed formulas are derived to compute V_I and I_T for the five studied space-filling curves.

A comprehensive study for the behavior of different space-filling curves with respect to irregularity is conducted. The scalability, fairness, and intentional bias of space-filling curves with respect to irregularity is analyzed. For scalability, we show that after a certain dimensionality and a certain grid size, the irregularity behavior

of all space-filling curves reaches its steady state value. Generally, the Peano SFC is more scalable than the other SFCs where the Peano SFC has the lowest irregularity for all dimensions and grid sizes. For fairness, we show that with the increase of dimensionality, the Hilbert and Gray SFCs are more fair than the others. For intensional bias, the Sweep and Scan SFCs have more intentional bias than other space-filling curves. Thus, the choice of a certain space-filling curve for a multi-dimensional application depends mainly on the nature of the application. For example, for an application that requires intentional bias toward one of the dimensions (e.g., real time applications), the Sweep or the Scan SFCs is the best choice. For high-dimensional applications (extracting features from multimedia applications), a scalable space-filling curve (e.g., the Peano SFC) is the best choice. A highlight of some practical applications for Network Attached Storage Devise (NASD) and multimedia-Aware disk scheduling is presented.

References

1. Abel, D.J., Mark, D.M.: A comparative analysis of some two-dimensional orderings. Int. J. Geogr. Inf. Syst. **4**(1), 21–31 (1990)
2. Abel, D.J., Smith, J.: A data structure and algorithm based on a linear key for a rectangle retrieval problem. Comput. Vis. Graph. Image Process. **24**, 1–13 (1983)
3. Alber, J., Niedermeier, R.: On multi-dimensional Hilbert indexing. In: International Computing and Combinatorics Conference, COCOON, Aug. 1998, pp. 329–338 (1998)
4. Aref, W.G., Kamel, I.: On multi-dimensional sorting orders. In: Proc. of the International Conference on Database and Expert Systems Applications, DEXA, Sept. 2000, pp. 774–783 (2000)
5. Aref, W.G., El-Bassyouni, K., Kamel, I., Mokbel, M.F.: Scalable QoS-aware disk-scheduling. In: International Database Engineering and Applications Symposium, IDEAS, July 2002
6. Asano, T., Ranjan, D., Roos, T., Welzl, E., Widmayer, P.: Space-filling curves and their use in the design of geometric data structures. Theor. Comput. Sci. **181**(1), 3–15 (1997)
7. Bartholdi, J.J., Platzman, L.K.: An O(n log n) traveling salesman heuristic based on space filling curves. Oper. Res. Lett. **1**(4), 121–125 (1982)
8. Bially, T.: Space-filling curves: their generation and their application to bandwidth reduction. IEEE Trans. Inf. Theory **15**(6), 658–664 (1969)
9. Bohm, C., Klump, G., Kriegel, H.-P.: XZ-Ordering: a space-filling curve for objects with spatial extension. In: Proceedings of the International Symposium on Advances in Spatial Databases, SSD, July 1999, pp. 75–90 (1999)
10. Breinholt, G., Schierz, C.: Algorithm 781: generating Hilbert's space-filling curve by recursion. ACM Trans. Math. Softw. **24**(2), 184–189 (1998)
11. Brinkhoff, T., Kriegel, H.-P., Seeger, B.: Efficient processing of spatial joins using R-trees. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD, May 1993, pp. 237–246 (1993)
12. Chen, S., Ooi, B.C., Tan, K.-L., Nascimento, M.A.: ST2B-tree: a self-tunable spatio-temporal B+-tree index for moving objects. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD, June 2008, pp. 29–42 (2008)
13. Cole, A.J.: A note on space filling curves. Softw. Pract. Exp. **13**(12), 1181–1189 (1983)
14. Comer, D.: The ubiquitous B-tree. ACM Comput. Surv. **11**(2), 121–137 (1979)
15. Faloutsos, C.: Gray codes for partial match and range queries. IEEE Trans. Softw. Eng. **14**(10), 1381–1393 (1988)
16. Faloutsos, C., Bhagwat, P.: Declustering using fractals. In: Proceedings of the International Conference on Parallel and Distributed Information Systems, Jan. 1993, pp. 18–25 (1993)
17. Faloutsos, C., Rong, Y.: DOT: a spatial access method using fractals. In: Proceedings of the IEEE International Conference on Data Engineering, ICDE, pp. 152–159 (1991)
18. Faloutsos, C., Roseman, S.: Fractals for secondary key retrieval. In: Proceedings of the ACM Symposium on Principles of Database Systems, PODS, pp. 247–252 (1989)

19. Finkel, R.A., Bentley, J.L.: Quad trees: a data structure for retrieval on composite keys. Acta Inform. **4**, 1–9 (1974)
20. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private queries in location-based services: anonymizers are not necessary. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD, June 2008, pp. 121–132 (2008)
21. Goldschlager, L.M.: Short algorithms for space-filling curves. Softw. Pract. Exp. **11**(1), 99–100 (1981)
22. Gray, F.: Pulse code communications. US Patent 2632058 (1953)
23. Hilbert, D.: Ueber stetige abbildung einer linie auf ein flashenstuck. Math. Ann. 459–460 (1891)
24. Jagadish, H.V.: Linear clustering of objects with multiple attributes. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD, June 1990, pp. 332–342 (1990)
25. Jensen, C.S., Tiesyte, D., Tradisauskas, N.: Robust B+-tree-based indexing of moving objects. In: Proceedings of the IEEE International Conference on Mobile Data Management, MDM, May 2006
26. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Trans. Knowl. Data Eng. **19**(12), 1719–1733 (2007)
27. Kamel, I., Faloutsos, C.: On packing R-trees. In: Proceedings of the ACM International Conference on Information and Knowledge Managemen, CIKM, Nov. 1993, pp. 490–499 (1993)
28. Kamel, I., Faloutsos, C.: Hilbert R-tree: An improved R-tree using fractals. In: Proceedings of the International Conference on Very Large Data Bases, VLDB, Sept. 1994, pp. 500–509 (1994)
29. Lawder, J.K., King, P.J.H.: Using space-filling curves for multi-dimensional indexing. In: Proceedings of the 17th British National Conference on Databases, BNCOD, July 2000, pp. 20–35 (2000)
30. Liao, S., Lopez, M.A., Leutenegger, S.: High dimensional similarity search with space-filling curves. In: Proceedings of the IEEE International Conference on Data Engineering, ICDE, Apr. 2001, pp. 615–622 (2001)
31. Mokbel, M.F., Aref, W.G.: Irregularity in multi-dimensional space-filling curves with applications in multimedia databases. In: Proceedings of the ACM International Conference on Information and Knowledge Managemen, CIKM, Nov. 2001, pp. 512–519 (2001)
32. Mokbel, M.F., Aref, W.G.: On query processing and optimality using spectral locality-preserving mappings. In: Proceedings of the International Symposium on Advances in Spatial and Temporal Databases, SSTD, July 2003
33. Mokbel, M.F., Aref, W.G., Kamel, I.: Performance of multi-dimensional space-filling curves. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS, Nov. 2002
34. Mokbel, M.F., Aref, W.G., Grama, A.: Spectral LPM: an optimal locality-preserving mapping using the spectral (not fractal) order. In: Proceedings of the IEEE International Conference on Data Engineering, ICDE, Mar. 2003
35. Mokbel, M.F., Aref, W.G., Kamel, I.: Analysis of multi-dimensional space-filling curves. GeoInformatica **7**(3), 179–209 (2003)
36. Mokbel, M.F., Aref, W.G., Elbassioni, K.M., Kamel, I.: Scalable multimedia disk scheduling. In: Proceedings of the IEEE International Conference on Data Engineering, ICDE, Mar. 2004
37. Moon, B., Jagadish, H., Faloutsos, C., Salz, J.: Analysis of the clustering properties of Hilbert space-filling curve. IEEE Trans. Knowl. Data Eng. **13**(1), 124–141 (2001)
38. Moore, E.H.: On certain crinkly curves. Trans. Am. Math. Soc. 72–90 (1900)
39. Morton, G.M.: A computer oriented geodetic data base and a new technique in file sequences. IBM (1966)
40. Orenstein, J.A.: Spatial query processing in an object-oriented database system. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD, May 1986, pp. 326–336 (1986)
41. Orenstein, J.A., Merrett, T.: A class of data structures for associative searching. In: Proceedings of the ACM Symposium on Principles of Database Systems, PODS, Apr. 1984, pp. 181–190 (1984)
42. Patrick, E.A., Anderson, D.R., Bechtel, F.K.: Mapping multidimensional space to one dimension for computer output display. IEEE Trans. Comput. **17**(10), 949–953 (1968)
43. Peano, G.: Sur une courbe qui remplit toute une air plaine. Math. Ann. **36**, 157–160 (1890)
44. Sevcik, K.C., Koudas, N.: Filter trees for managing spatial data over a range of size granularities. In: Proceedings of the International Conference on Very Large Data Bases, VLDB, Sept. 1996, pp. 16–27 (1996)
45. Shepherd, J., Zhu, X., Megiddo, N.: A fast indexing method for multidimensional nearest neighbor search. SPIE, Storage Retr. Image Video Databases **3656**, 350–355 (1998)
46. Thottethodi, M., Chatterjee, S., Lebeck, A.: Tuning Strassen matrix multiplication algorithm for memory efficiency. In: Proceedings High Performance Computing ad Networking, SC, Nov. 1998

47. Tropf, H., Herzog, H.: Multidimensional range search in dynamically balanced trees. Angew. Inform., 71–77 (1981)
48. Velho, L., Gomes, J.: Stochastic screening dithering with adaptive clustering. In: Proceedings of the ACM Conference on Computer Graphics, pp. 273–276 (1995)
49. White, M.: N-Trees: Large ordered indexes for multi-dimensional space. Statistical research division. US Bureau of the Census (1980)
50. Witten, I.H., Neal, M.: Using Peano curves for bilevel display of continuous tone images. IEEE Comput. Graph. Appl., 47–52 (1982)
51. Witten, I.H., Wyvill, B.: On the generation and use of space-filling curves. Softw. Pract. Exp. **3**, 519–525 (1983)