Irregularity in Multi-Dimensional Space-Filling Curves with Applications in Multimedia Databases

Mohamed F. Mokbel
Walid G. Aref

Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398
{mokbel,aref}@cs.purdue.edu

ABSTRACT

A space-filling curve is a way of mapping the multi-dimensional space into the one-dimensional space. It acts like a thread that passes through every cell element (or pixel) in the N-dimensional space so that every cell is visited at least once. Thus, a space-filling curve imposes a linear order of the cells in the N-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 a brief knowledge of the mapping scheme provided by each space-filling curve. Irregularity is proposed as a quantitative measure of the quality of the mapping of the space-filling curve. Closed formulas are developed to compute the irregularity for any general dimension D with N points in each dimension for different space-filling curves. A comparative study of different space-filling curves with respect to irregularity is conducted and results are presented and discussed. The applicability of this research in the area of multimedia databases is illustrated with a discussion of the problems that arise.

1. INTRODUCTION

Mapping the multi-dimensional space into the one-dimensional domain plays an important role in every application that involves multi-dimensional data. Multimedia databases, Geographic Information Systems (GIS), QoS routing and Image processing are examples of multi-dimensional applications. Modules that are commonly used in multi-dimensional applications include searching, sorting, scheduling, spatial access methods, indexing and clustering. Numerous research has been conducted for developing efficient algorithms and data structures for these modules for one-dimensional data. In most cases, modifying the existing one-dimensional algorithms and data structures to deal with multi-dimensional data results in spaghetti-like programs to handle many special cases. The cost of maintaining and developing such code degrades the system performance.

Mapping from the multi-dimensional space into the one-dimensional domain provides a pre-processing step for multi-dimensional applications. The pre-processing step takes the multi-dimensional data as input and outputs the same set of data represented in the one-dimensional domain. The idea is to keep the existing algorithms and data structures independent of the dimensionality of data. The objective of the mapping is to represent a point from the D-dimensional space by a single integer value that reflects the various dimensions of the original space. Such a mapping is called a locality-preserving mapping in the sense that, if two points are near to each other in the D-dimensional space, then they will be near to each other in the one-dimensional space. An ideal mapping maps each multi-dimensional interval into a one-dimensional interval that contains precisely the image of the corresponding interval.

Space-Filling Curves (SFCs) have been extensively used as a mapping scheme from the multi-dimensional space into the one-dimensional space. A space-filling curve is a thread that goes through all the points in the space while visiting each point only one time. Thus, a space-filling curve imposes a linear order of points in the multi-dimensional space. Space-Filling curves are discovered by Peano [41] where he introduces a mapping from the unit interval to the unit square. Hilbert [24] generalizes the idea to a mapping of the whole space. Following Peano and Hilbert curves, many space-filling curves are proposed , e.g., [8, 44]. Examples of space-filling curves are given in Figure 1. According to the classification in [8], space-filling curves are classified into two categories: recursive space-filling curves (RSFC) and non-recursive space-filling curves. An RSFC is an SFC that can be recursively divided into four square RSFCs of equal size. Examples of RSFCs are the Peano SFC (Figure 1c), the Gray SFC (Figure 1d) and the Hilbert SFC (Figure 1e). For a historical survey and more types of space-filling curves, the reader is referred to [42].

With the variety of space-filling curves and the wide spread of multi-dimensional applications, the selection of the appropriate space-filling curve for a certain application is not a trivial task. One way is to perform many simulation experiments over different space-filling curves. However, this is not practical in terms of execution time. Another way is to tailor a new space-filling curve for each application, e.g., as in [8, 9, 36]. However, with the increase of multi-dimensional applications, it becomes a hard task to tailor a new space-filling curve for each application.

The objective of this paper is to provide a systematic and a scalable framework for selecting the appropriate space-

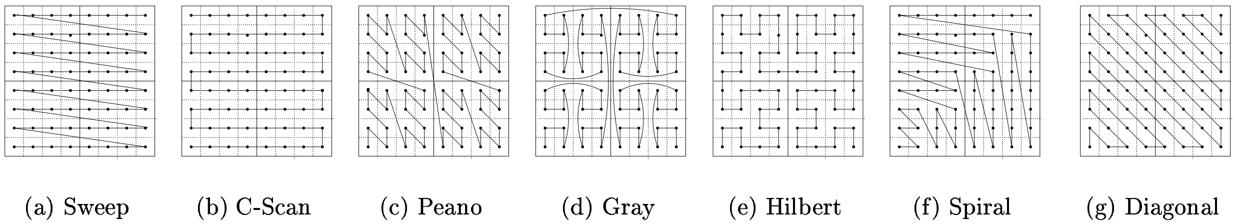


Figure 1: 2D Space-Filling Curves.

filling curve for any required application. To achieve this goal, we go through three main steps. First, we introduce the notion of *irregularity* as a measure of goodness for space-filling curves. Second, we analyze the most commonly used space-filling curves, namely, the Peano (Figure 1c), the Gray (Figure 1d) and the Hilbert (Figure 1e) SFCs as case studies. Finally, we develop closed formulas to compute the irregularity of each space-filling curve for any general dimension k with grid size N points in each dimension. For more measures of goodness and more case studies the reader is referred to [7].

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 introduces irregularity as a measure of goodness of space-filling curves. Section 4 analyzes the Peano, Gray and Hilbert SFCs and develops closed formulas to compute the irregularity of each space-filling curve. Section 5 conducts comprehensive comparison among different space-filling curves. In Section 6, we discuss the applicability of this research in multimedia databases. Finally, Section 7 concludes the paper and points out future work.

2. RELATED WORK

Although space-filling curves were discovered in the last century [24, 34, 41], 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 by [38] for spatial join of multi-dimensional data. The multi-dimensional data is transformed into the one-dimensional domain using Z-order [39], which is the same as the Peano SFC [41]. The transformed data is stored in a one-dimensional data structure, the B^+-Tree [14], and a spatial join algorithm is applied. The Gray [22] and the Hilbert [24] space-filling curves are used for answering range queries in [15, 25], respectively. [17, 18] use space-filling curves as a spatial access method where the multi-dimensional data is stored in one-dimensional media (disk) using the Hilbert mapping. This achieves clustering and hence reduces the number of disk accesses and improves the response time. [26] uses the Hilbert SFC in packing the R-Tree [23], where a set of rectangles are sorted according to the Hilbert order, and then are packed into the R-Tree nodes. Similar ideas for constructing R-trees using space-filling curves are proposed in [27]. The Z-order [39] (Peano SFC [41]) is used in [11] as a spatial access method to enhance the performance of spatial join. 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 performed in [43] based on the Hilbert SFC. The Hilbert SFC is also used in multi-dimensional indexing in [29, 30] and for answering nearest-neighbor queries in [31].

The clustering properties of space-filling curves make it suitable for memory management. [52] proposes the use of the Hilbert SFC for pixel traversal in the image plane for the distributed shared volume buffer (DSVB) environment [52]. Old methods for pixel traversal rely on one-way traversal (the Sweep SFC) and two way traversal (the C-Scan SFC). Using the Hilbert SFC in pixel traversal achieves better performance in cache, thus reducing the time of remote data fetching through the network. [45] uses space-filling curves to improve memory performance for dense matrix multiplication. The matrix elements are ordered according to the Morton-order [35] (the Peano SFC) rather than the usual row-major or column-major order. This improves the cache performance of Strassan's multiplication algorithm. [32] explores the use of the Hilbert SFC for improving memory management for irregular applications.

Space-Filling Curves are used in image processing applications. In digital halftoning, [50] improves the error diffusion by replacing the left-to-right row-by-row order of processing pixels by an order imposed by the Peano SFC. [13] generalizes the Peano SFC for non-square grids to produce Murray polygons [13] to use it in digital halftoning. Other algorithms for digital halftoning that are based on Hilbert SFC are proposed in [47, 48, 53].

Other uses of space-filling curve include data-parallel applications [40], parallel algorithms for mesh-connected networks [37], and disk scheduling [5]. Some applications need a tailored space-filling curve. In [8], a new recursive space-filling curve is proposed that guarantees an upper bound of three seek operations to any 2D square query. In [36], an H-index ordering is proposed for mesh-indexing. XZ-ordering is proposed by [9] to map objects with spatial extension. The XZ-order is an extension of the Z-order by extending each region in Z-order by a factor of two in each dimension.

In [4], the notion of Hilbert indexing is generalized to arbitrary dimensions. The Hilbert SFC is structurally analyzed, which helps in understanding how the Hilbert SFC is built in the multi-dimensional space. [5] studies the properties of several space filling curves in the two- and three-dimensional spaces, and introduces new measures to describe the behavior of any space-filling curve. In [33], the clustering properties of the Hilbert curve is analyzed by deriving closed formulas for the number of clusters in a given query region.

Numerous algorithms are developed for efficiently generating different space-filling curves. Recursive algorithms for generating the Hilbert SFC are proposed in [10, 12, 21, 51] and for the Peano SFC in [12, 51]. A table-driven algorithm

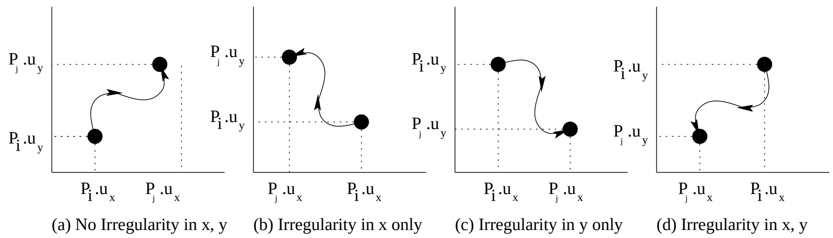


Figure 2: Irregularity in 2D space.

for the Peano and Hilbert SFCs is proposed in [21]. An algorithm for computing the order of any point in the Hilbert, Peano, and Gray SFCs is proposed in [18]. For a comparison of different space-filling curves, a reader is referred to [1, 5, 7, 16, 42].

3. IRREGULARITY IN SPACE-FILLING CURVES

An optimal space-filling curve is one that sorts points in space in ascending order for all dimensions. 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 is measured for each dimension separately. It gives an indicator of how a space-filling curve is far from the optimal. The lower the irregularity, the better the space-filling curve.

Definition 1. For any two points, say P_i and P_j, in the D-dimensional space with coordinates $(P_i.u_1, P_i.u_2, \ldots, P_i.u_D), (P_j.u_1, P_j.u_2, \ldots, P_j.u_D)$, respectively, and for a given space-filling curve, if P_i is visited before P_j, we say that an irregularity occurs between P_i and P_j in dimension k iff $P_j.u_k < P_i.u_k$. In other words, we say that (P_i, P_j) is an irregular tuple.

Figure 2 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.

Formally, for a given space-filling curve in the D-dimensional space with grid size N per dimension, the number of irregularities for any dimension k is:

$$I(k, N, D) = \sum_{j=1}^{N^D} \sum_{i=1}^{j-1} f_{ij} \quad where \quad f_{ij} = 1 \quad iff \quad P_i.u_k > P_j.u_k$$

An optimal space-filling curve for any dimension k would have no irregularity, i.e., $I_{opt}(k, N, D) = 0$. In contrast, the worst-case space-filling curve for any dimension k is to sort all points in reverse order with respect to k.

LEMMA 1. *In a D-dimensional space with grid size N, the total number of irregularities in the worst-case space-filling curve along any dimension k is:*

$$I_{wr}(k, N, D) = \frac{1}{2} N^{2D-1}(N-1)$$

PROOF. The proof is omitted for brevity. The reader is referred to [7] for a detailed proof. □

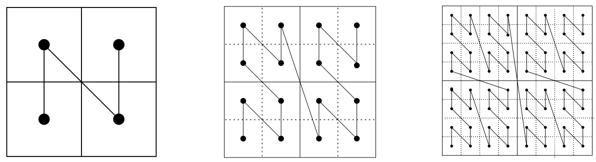
4. MATHEMATICAL ANALYSIS OF SPACE-FILLING CURVES

The time complexity for calculating the irregularity is $O(N^{2D})$ where D is the number of dimensions and N is the number of points in each dimension. Consider the case of 20 dimensions with 16 points in each dimension, we need 16^{40} operations to compute the irregularity of a space-filling curve. To avoid this excessive operation, in this section we derive closed formulas that compute the irregularity for any dimension k in a D-dimensional space with N points in each dimension. In this paper, we concentrate only on the recursive space-filling curves: Peano, Gray, and Hilbert SFCs. For closed formulas of other space-filling curves and the proofs of all the equations presented in this section, the reader is referred to [7].

4.1 The Peano SFC

The Peano SFC (Figure 1c) is introduced by Peano [41] and is also called Morton encoding [35], quad code [19], bit-interleaving [46], N-order [49], locational code [2], or Z-order [39]. The Peano SFC is constructed recursively as in Figure 3. The basic step (Figure 3a) 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 and the least significant digit is represented by its y position. The Peano SFC orders these points in ascending order (00, 01, 10, 11). Figure 3b contains four repeated blocks of Figure 3a at a finer resolution and is visited in the same order as in Figure 3a. Similarly, Figure 3c contains four repeated blocks of Figure 3b at a finer resolution.

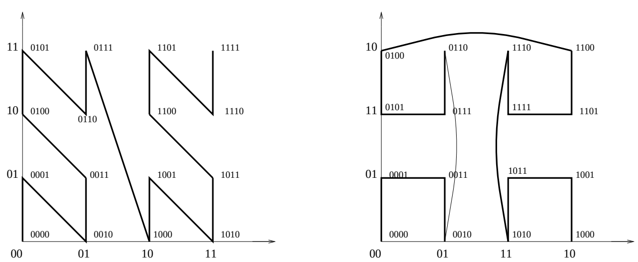
To extend the Peano SFC to the multi-dimensional space, we present the idea of bit-interleaving in the two-dimensional space as shown in Figure 4a. Each point in the space takes a binary number that results from interleaving bits of the two dimensions. The bits are interleaved according to the interleaving vector (0,1,0,1). This corresponds to taking the first and third bits from dimension 0 (x) and taking the second and fourth bits from dimension 1 (y). For a D-dimensional space with four points in each dimension, the interleaving vector is $(0, 1, 2, \ldots, D-1, 0, 1, 2, \ldots, D-1)$. For grid size N points in each dimension, the term $0, 1, 2, \ldots, D-1$ is repeated $LogN$ times.



(a) (b) (c)

Figure 3: The Peano Recursive SFC.

The points are visited in ascending order according to their binary number representation. Table 1 gives an example of computing the Peano order for two- and three-dimensional points with a grid size of eight points in each dimension.



(a) Peano (b) Gray

Figure 4: Bit-Interleaving in Peano/Gray SFC.

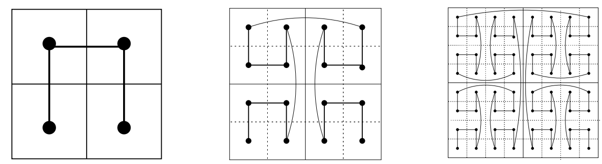
LEMMA 2. *In a D-dimensional space with grid size N, the number of irregularities in any dimension k for the Peano SFC is:*

$$I_P(k,N,D) = \frac{N^D\left(N^D-1\right)\left(2^D-2^{D-k-1}-1\right)}{4(2^D-1)} - \frac{N^D\left(N^{D-1}-1\right)}{4}$$

PROOF. The proof is omitted for brevity. The reader is referred to [7] for a detailed proof. □

4.2 The Gray SFC

The Gray SFC (Figure 1d) uses the Gray code representation [22] in contrast to the binary code representation as in the Peano SFC. Figure 5 gives the recursive construction of the Gray SFC. The basic step (Figure 5a) contains four points in the four quadrants of the space. As in the Peano SFC, each quadrant is represented by two binary digits. The most significant digit is represented by its x position and the least significant digit is represented by its y position. The Gray SFC orders these points in ascending order according to the Gray code (00, 01, 11, 10). Figure 5b contains four repeated blocks of Figure 5a at a finer resolution and is visited in Gray order. Unlike the Peano SFC, the first and the fourth blocks have the same orientation as those of Figure 5a, while the second and the third blocks are constructed by rotating the block of Figure 5a by 180^0. Similarly, Figure 5c is constructed from two blocks of Figure 5b at a finer resolution and two blocks of the rotation of Figure 5b by 180^0.



(a) (b) (c)

Figure 5: The Gray Recursive SFC.

To extend the Gray SFC to the multi-dimensional space, we use the same idea of bit-interleaving as in the Peano SFC. Figure 4b gives the bit-interleaving in the two-dimensional space with four points in each dimension. Table 2 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 3. *In a D-dimensional space with grid size N, the number of irregularities in any dimension k for the Gray SFC is:*

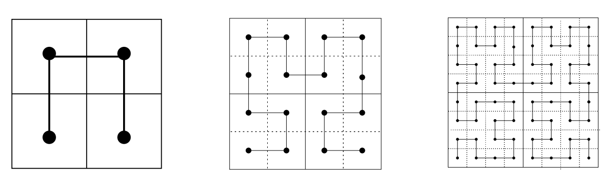
$$I(0,N,D) = \frac{N^{2D-1}}{4}\left(\frac{N}{2}-1\right)$$

$$I(k,N,D) = \frac{N^{2D-1}}{4}(N-1) \qquad k>0$$

PROOF. The proof is omitted for brevity. The reader is referred to [7] for a detailed proof. □

4.3 The Hilbert SFC

Figure 6 gives the recursive construction of the Hilbert SFC. The basic block of the Hilbert SFC (Figure 6a) is the same as the basic block of the Gray SFC (Figure 5a). The basic block is repeated four times at a finer resolution in the four quadrants, as given in Figure 6b. The quadrants are visited in their gray order. The second and third blocks in Figure 6b have the same orientation as in Figure 6a. The first block is constructed from rotating the block of Figure 6a by 90^0, while the fourth block is constructed by rotating the block of Figure 6 by -90^0. Figure 6c is constructed from Figure 6b in an analogous manner.



(a) (b) (c)

Figure 6: The Hilbert Recursive SFC.

Point	Dimensions		Bit	Decimal	Point	Dimensions			Bit	Decimal
	0	1	Interleaving	Order		0	1	2	Interleaving	Order
(2,1)	**010**	001	001001	9	(0,1,3)	**000**	*001*	011	000001011	11
(5,3)	**101**	011	100111	39	(2,1,4)	**010**	*001*	100	001100010	98
(7,0)	**111**	000	101010	42	(7,0,7)	**111**	*000*	111	101101101	365

Table 1: An Example of two- and three-dimensional Peano orders with grid size 8 in each dimension.

Point	Dimensions		Bit	Decimal	Point	Dimensions			Bit	Decimal
	0	1	Interleaving	Order		0	1	2	Interleaving	Order
(2,1)	**011**	001	001011	13	(0,1,3)	**000**	*001*	010	000001010	12
(5,3)	**111**	010	101110	52	(2,1,4)	**011**	*001*	110	001101110	75
(7,0)	**100**	000	100000	63	(7,0,7)	**100**	*000*	100	100000100	384

Table 2: An Example of two- and three-dimensional Gray orders with grid size 8 in each dimension.

LEMMA 4. *In a D-dimensional space with grid size N, the number of irregularities in any dimension k for the Hilbert SFC is:*

$$I(k^*, N, D) = \frac{N(2^{2D-2}-1)}{4}\left(\frac{N^{2D-1}-1}{2^{2D-1}-1} - \frac{N^{2D-2}-1}{2^{2D-2}-1}\right)$$

$$I(k, N, D) = \frac{N^{2D-1}}{4}(N-1) \qquad k \neq k^*$$

where $k^* = (3D + Log(N) - 5)\ mod\ D$.

PROOF. The proof is omitted for brevity. The reader is referred to [7] for a detailed proof. □

5. PERFORMANCE EVALUATION

In this section, we use the irregularity as our measure of performance. The irregularity of the Peano, Gray, and Hilbert SFCs is calculated from Lemmas 2, 3, and 4, respectively. For comparison purposes, we use the worst-case irregularity (Lemma 1) as our base point. The number of irregularities in a space-filling curve along any dimension is presented as a ratio of the number of irregularities in the worst-case scenario. In the following sections, we conduct three sets of experiments. First, in Section 5.1, we measure the scalability of space-filling curves for high-dimensional spaces. In Section 5.2, we measure the fairness of the different space-filling curves[1]. Finally, in Section 5.3, we discuss the intentional bias of the different space-filling curves[2].

5.1 Scalability of Space-Filling Curves

In this section, we address the issue of scalability of SFC-based applications, e.g., when the number of dimensions increases or when the number of points per dimension increases. As defined in Section 3, the irregularity is calculated for each dimension individually. To measure the performance of a space-filling curve with respect to irregularity, we calculate the average number of irregularities for all dimensions.

In Figure 7, we measure the mean irregularity for up to 12 dimensions with a grid size of 16 points in each dimension. The Peano SFC gives the best results (less irregularity) while the Gray and Hilbert SFCs have the same performance, except for the two-dimensional case. The irregularity of the Peano SFC ranges from 22-45%. For the Gray and Hilbert SFCs, the irregularity ranges from 35-50%. Figure 8 gives a comparison of the irregularity of space-filling curves in the four-dimensional space, while the grid size (the number of points) varies from 4 to 256. For gird sizes greater than 16, all space-filling curves tend to exhibit constant behavior. The Peano SFC has better irregularity in the range 33-38%, while both the Gray and Hilbert SFCs have irregularity in the range 41-44%.

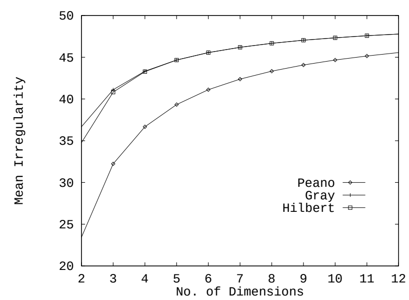


Figure 7: The mean irregularity of the three SFCs for a grid size of 16 points per dimension.

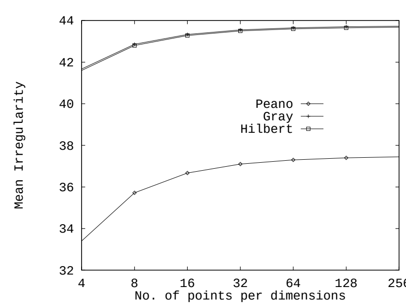


Figure 8: The mean irregularity of the three SFCs in the four-dimensional space.

5.2 Fairness of Space-Filling Curves

A very critical point for SFC-based applications is how to assign the different parameters to the dimensions of the space-filling curve. We say that a space-filling curve is fair if it results in similar irregularity for all dimensions. In this section, we use the standard deviation of irregularity over all the dimensions as a measure of fairness of the space-filling curves where a low standard deviation indicates more

[1] We say that a space-filling curve is fair if it has similar behavior towards all dimensions.

[2] We say that a space-filling curve is intentionally biased toward dimension k, if it has less irregularity in dimension k with respect to all other dimensions.

fairness.

In Figure 9, we vary the space dimensionality from 2 to 12 with 16 points in each dimension. For less than nine dimensions, the Peano SFC gives the fairest behavior. For more than nine dimensions, the Gray and Hilbert SFCs exhibit more fairness than the Peano SFC. In Figure 10, we apply the same experiment in the four-dimensional space while varying the grid size (the number of points) in each dimension from 4 to 256. The Peano SFC always gives the best results.

5.3 Intentional Bias of Space-Filling Curves

Some applications may have only one important dimension, while the other dimensions are not with the same significant importance. As a result, we need a space-filling curve that is biased to some dimensions. We say that a space-filling curve is biased to dimension k if it results in low irregularity in k relative to the other dimensions. For such applications, we develop the experiment given in Figure 11. Although we run the experiment in a four-dimensional space, we plot only one dimension. The plotted dimension is the most favored dimension for each space-filling curve, e.g., the one with the lowest irregularity. Figure 11 shows that the favored dimension of the Gray SFC has constant irregularity regardless of the number of the dimensions used. For the Hilbert SFC, the favored dimension has less irregularity than the Gray SFC for less than six dimensions, and then it exhibits the same performance as the Gray SFC. Similarly, the Peano SFC has much lower irregularity for less than nine dimensions, and then it exhibits the same performance as in the Gray and Hilbert SFCs.

Figures 12, 13 and 14 give the irregularity for each of the four dimensions of the Peano, Gray, and Hilbert SFCs, respectively. For the Peano SFC (Figure 12), 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 (Figure 13), namely, the first dimension. The Gray SFC deals with the remaining dimensions fairly though with high irregularity. All dimensions except the first have 50% irregularity. Figure 14 gives an interesting property of the Hilbert SFC. As in the Gray SFC, at any point, the Hilbert SFC has only one favored dimension while all the other dimensions have 50% irregularity. However, the favored dimension of the Hilbert SFC changes in a round robin fashion. The Hilbert SFC favors the third dimension at the grid size of four points per dimension (i.e., N=4). At grid size eight points per dimension (i.e., N=8), the Hilbert SFC favors the fourth dimension. The bold solid line in Figure 14 represents the favored dimension and it has the same performance as the Gray SFC.

6. APPLICATIONS OF SPACE-FILLING CURVES IN MULTIMEDIA DATABASES

In this section, we investigate the applicability of this research in multimedia databases as one of the multi-dimensional applications. We emphasize on the use of space-filling curve mapping in network-attached storage devices (NASDs) and QoS-aware disk scheduling.

Network-Attached Storage Devices NASDs. Writing efficient schedulers is becoming a very challenging task, given the increase in demand of such systems. Consider, the case of network-attached storage devices (NASDs) [20, 28] as a building block for a multimedia server (e.g., see [3]). NASDs are smart disks that are attached directly to the network. In a multimedia server, a major part of a NASD function goes towards fulfilling the real-time requests of users. This involves disk and network scheduling with real-time constraints, possibly with additional requirements like request priorities, and quality-of-service guarantees. NASDs requirements can be mapped in the multi-dimensional space and a SFC-based scheduler is used. The type of space-filling curve used in NASD scheduling is determined by its requirements. For example, in NASD, we might be interested in reducing the number of requests that lose their deadlines, more than we are in increasing the disk or network bandwidth. In this case, we may want to favor the real-time deadline dimension of the scheduling space. As a result, a space-filling curve with intentional bias is favored.

QoS-Aware Disk Scheduling. Consider the problem of disk scheduling in multimedia servers. 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. Writing a disk scheduler that handles real-time and QoS constraints in addition to maximizing the disk bandwidth is challenging and a hard task [6]. Scheduler parameters can be mapped to space dimensions and an SFC-based scheduler is used.

7. CONCLUSION

Multi-dimensional scheduling boils down to finding a linear order for a set of points that lie in the D-dimensional space. Space-filling curves map the multi-dimensional space into one-dimensional space. The notion of irregularity is proposed as a measure of goodness for the mapping of each space-filling curve. The characteristics of the Peano, Gray, and Hilbert SFCs are analyzed. Closed formulas for the number of irregularities in any dimension k of the D-dimensional space are derived for the Peano, Gray, and Hilbert SFCs. Experiments are conducted that show how each of the space-filling curves varies from the other curves with respect to irregularity.

8. REFERENCES

[1] D. J. Abel and D. M. Mark. A comparative analysis of some two-dimensional orderings. *Intl. Journal of Geographical Information Systems*, 4(1):21–31, 1990.

[2] D. J. Abel and J. Smith. A data structure and algorithm based on a linear key for a rectangle retrieval problem. *Computer Vision Graphics Image Processing*, 24:1–13, 1983.

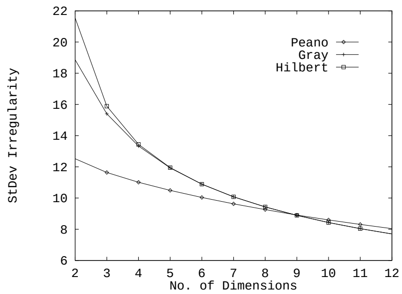


Figure 9: Standard deviation of the irregularity for grid size 16 per dimension.

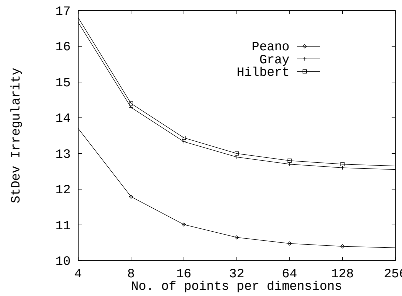


Figure 10: Standard deviation of the irregularity for four-dimensional space.

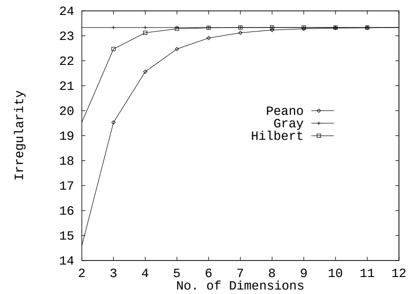


Figure 11: Irregularity for the favored dimension for a grid size 16 per dimension.

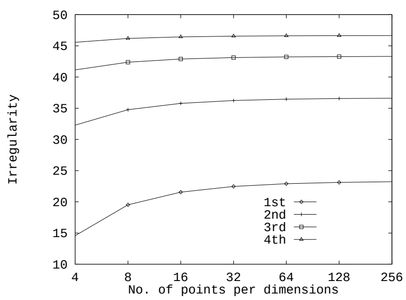


Figure 12: Irregularity for all dimensions for the Peano SFC.

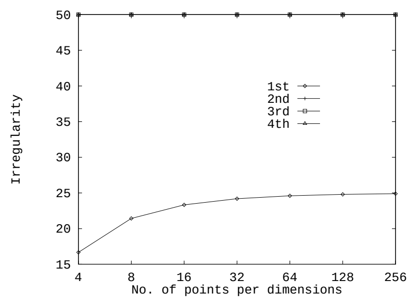


Figure 13: Irregularity for all dimensions for the Gray SFC.

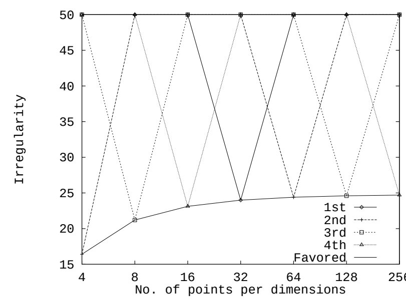


Figure 14: Irregularity for all dimensions for the Hilbert SFC.

[3] C. Akinlar, W. G. Aref, I. Kamel, and S. Mukherjee. Automatic disks: The building block for a scalable distributed file system. In *In Proc. of the 5th Intl. Workshop on Multimedia Information Systems, MIS*, Palm Springs Desert, Oct. 1999.

[4] J. Alber and R. Niedermeier. On multi-dimensional hilbert indexing. In *In Proc. of the 4th Intl. Computing and Combinatorics Conference, COCOON*, pages 329–338, Taipei, Taiwan, Aug. 1998.

[5] W. G. Aref and I. Kamel. On multi-dimensional sorting orders. In *In Proc. of the 11th Intl. Conf. on Database and Expert Systems Applications, DEXA*, pages 774–783, London, Sept. 2000.

[6] W. G. Aref, I. Kamel, and S. Ghandeharizadeh. Disk scheduling in video editing systems. *IEEE Trans. on Knowledge and Data Engineering, TKDE, To appear*, 2001.

[7] W. G. Aref, M. F. Mokbel, and I. Kamel. On the analysis of high-dimensional space-filling curves. Technical report, Computer Sciences Department, Purdue University, IN, Mar. 2001.

[8] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer. Space-filling curves and their use in the design of geometric data structures. *Theoretical Computer Science, TCS*, 181(1):3–15, 1997.

[9] C. Bohm, G. Klump, and H.-P. Kriegel. *xz*-ordering: A space-filling curve for objects with spatial extensio. In *In Proc. of 6th Intl. Symp. on Large Spatial Databases, SSD*, pages 75–90, Hong Kong, July 1999.

[10] G. Breinholt and C. Schierz. Algorithm 781: Generating hilbert's space-filling curve by recursion. *ACM Trans. on Mathematical Software, TOMS*, 24(2):184–189, June 1998.

[11] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joints using r-trees. In *In Proc. of the intl. conf. on Management of data, SIGMOD*, pages 237–246, Washington D.C., May 1993.

[12] A. J. Cole. A note on space filling curves. *Software–Practice and Experience, SPE*, 13(12):1181–1189, 1983.

[13] A. J. Cole. Halftoning without dither or edge enhancement. *The Visual Computer*, 7:232–246, 1991.

[14] D. Comer. The ubiquitous b-tree. *ACM Computing Surveys*, 11(2):121–137, June 1979.

[15] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Trans. on Software Engineering, TSE*, 14(10):1381–1393, Oct. 1988.

[16] C. Faloutsos. Analytical results on the quadtree decomposition of arbitrary rectangles. *Pattern Recognition Letters*, 13(1):31–40, Jan. 1992.

[17] C. Faloutsos and Y. Rong. Dot: A spatial access method using fractals. In *In Proc. of Intl. Conf. on Data Engineering, ICDE*, pages 152–159, Kobe, Japan, Apr. 1991.

[18] C. Faloutsos and S. Roseman. Fractals for secondary key retrieval. In *In Proc. of the 8th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, PODS*, pages 247–252, Philadelphia, Mar. 1989.

[19] R. A. Finkel and J. L. Bently. Quad trees: a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[20] G. Gibson, D. Nagle, K. Amiri, J. Butler, F. W.

Chang, H. Gobioff, C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka. File server scaling with network-attached secure disks. In *In Proc. of Intl. Conf. on Measurement and Modeling of Computer Systems, SIGMETRICS*, pages 272–284, Seatle, Washington, June 1997.

[21] L. M. Goldschlager. Short algorithms for space-filling curves. *Software–Practice and Experience, SPE*, 11(1):99–100, 1981.

[22] F. Gray. Pulse code communications. *US Patent 2632058*, 1953.

[23] A. Guttman. R-trees: A dynamic index structure for spatial indexing. In *In Proc. of the intl. conf. on Management of data, SIGMOD*, pages 47–57, Boston, MA, June 1984.

[24] D. Hilbert. Ueber stetige abbildung einer linie auf ein flashenstuck. *Mathematishe Annalen*, pages 459–460, 1891.

[25] H. V. Jagadish. Linear clustering of objects with multiple attributes. In *In Proc. of the intl. conf. on Management of data, SIGMOD*, pages 332–342, Atlantic City, NJ, June 1990.

[26] I. Kamel and C. Faloutsos. On packing r-trees. In *In Proc. of the 2nd Intl. Conf. on Information and knowledge Management, CIKM*, pages 490–499, Washington D. C., Nov. 1993.

[27] I. Kamel and C. Faloutsos. Hilbert r-tree: An improved r-tree using fractals. In *In Proc. of the 20th Intl. Conf. on Very Large Data Bases, VLDB*, pages 500–509, Santiago, Chile, Sept. 1994.

[28] R. H. Katz. High performance network- and channel-attached storage. *Proceedings of IEEE*, 80(8), Aug. 1992.

[29] J. K. Lawder and P. J. H. King. Using space-filling curves for multi-dimensional indexing. In *In Proc. of the 17th British National Conf. on Databases, BNCOD*, pages 20–35, UK, July 2000.

[30] J. K. Lawder and P. J. H. King. Querying multi-dimensional data indexed using the hilbert space filling curve. *SIGMOD Record*, 30(1), Mar. 2001.

[31] S. Liao, M. A. Lopez, and S. Leutenegger. High dimensional similarity search with space-filling curves. In *In Proc. of Intl. Conf. on Data Engineering, ICDE*, pages 615–622, Heidelberg, Germany, Apr. 2001.

[32] J. M. Mellor-Crummey, D. B. Whalley, and K. Kennedy. Improving memory hierarchy performance for irregular applications. In *In Proc. of the Intl. Conf. on Supercomputing, ICS*, pages 425–433, Rhodes, Greece, June 1999.

[33] B. Moon, H. Jagadish, C. Faloutsos, and J. Salz. Analysis of the clustering properties of hilbert space-filling curve. *IEEE Trans. on Knowledge and Data Engineering, TKDE*, 13(1):124–141, 2001.

[34] E. H. Moore. On certain crinkly curves. *Trans. Am. Math Soc.*, pages 72–90, 1900.

[35] G. Morton. A computer oriented geodetic data base and a new technique in file sequences. *IBM*, 1966.

[36] R. Niedermeier, K. Reinhardt, and P. Sanders. Towards optimal locality in mesh-indexing. In *In Proc. of the 11th Intl. Symp. on Fundamentals of Computation Theory, FCT*, pages 364–375, Krakow, Poland, Sept. 1997.

[37] R. Niedermeier and P. Sanders. On the manhattan-distance between points on space-filling mesh-indexing. Technical Report IB 18/96, Universitat Karlsruhe, Fakultat fur Informatik, May 1996.

[38] J. A. Orenstein. Spatial query processing in an object-oriented database system. In *In Proc. of the intl. conf. on Management of data, SIGMOD*, pages 326–336, Washington D.C., May 1986.

[39] J. A. Orenstein and T. Merrett. A class of data structures for associative searching. In *In Proc. of the 3rd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, PODS*, pages 181–190, Ontario, Canada, Apr. 1984.

[40] C.-W. Ou, M. Gunwani, and S. Ranka. Architecture-independent locality-improving transformations of computational graphs embedded in k-dimensions. In *In Proc. of the 9th ACM Intl. Conf. on Supercomputing, ICS*, pages 289–298, Barcelona, Spain, July 1995.

[41] G. Peano. Sur une courbe qui remplit toute une air plaine. *Mathematishe Annalen*, 36:157–160, 1890.

[42] H. Sagan. *Space Filling Curves.* Springer, Berlin, 1994.

[43] K. C. Sevcik and N. Koudas. Filter trees for managing spatial data over a range of size granularities. In *In Proc. of the 22th Intl. Conf. on Very Large Data Bases, VLDB*, pages 16–27, Bombay, India, Sept. 1996.

[44] W. Sierpinski. Sur une nouvelle courbe qui remplit toute une aire plaine. *Bull. Acad. Sci. Cracovie, SerieA*, pages 462–478, 1912.

[45] M. Thottethodi, S.Chatterjee, and A. Lebeck. Tuning strasse matrix multiplication algorithm for memory efficiency. In *In Proc. of SC98: High Performance Computing ad Networking*, Orlando, FL, Nov. 1998.

[46] H. Tropf and H.Herzog. Multidimensional range search in dynamically balanced trees. *Angewandte Informatik*, pages 71–77, 1981.

[47] L. Velho and J. Gomes. Digital halftoning with space filling curves. *Computer Graphics*, 25(4):81–90, July 1991.

[48] L. Velho and J. Gomes. Stochastic screening dithering with adaptive clustering. In *In Proc. of the 22nd ACM conf. on Computer graphics*, pages 273–276, 1995.

[49] M. White. N-trees: Large ordered indexes for multi-dimensional space. statistical research division. *US Bureau of the Census*, 1980.

[50] I. Witten and M. Neal. Using peano curves for bilevel display of continuous tone images. *IEEE Computer Graphics and Applications*, pages 47–52, 1982.

[51] I. Witten and B. Wyvill. On the generation and use of space-filling curves. *Software–Practice and Experience*, 3:519–525, 1983.

[52] H. Zhang and S. Liu. Order of pixel traversal and parallel volume ray-tracing on the distributed shared volume buffer. *Eurographics Workshop on Visualization*, 1995.

[53] Y. Zhang and R. E. Webber. Space diffusion: An improved parallel halftoning technique using space-filling curves. *In Computer Graphics Proc.*, pages 305–312, Aug. 1993.