Performance of Multi-Dimensional Space-Filling Curves*

Mohamed F. Mokbel
Department of Computer Sciences, Purdue University
West Lafayette, IN
mokbel@cs.purdue.edu

Walid G. Aref
Department of Computer Sciences, Purdue University
West Lafayette, IN
aref@cs.purdue.edu

Ibrahim Kamel
Panasonic Information and Networking Technologies Laboratory, Princeton, NJ
ibrahim@research.panasonic.com

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 D-dimensional space so that every cell is visited exactly once. 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. A space-filling curve consists of a set of segments. Each segment connects two consecutive multi-dimensional points. Five different types of segments are distinguished, namely, *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still*. A description vector $V = (J, C, R, F, S)$, where J, C, R, F, and S, are the percentages of *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still* segments in the space-filling curve, encapsulates all the properties of a space-filling curve. The knowledge of V facilitates the process of selecting the appropriate space-filling curve for different applications. Closed formulas are developed to compute the description vector V for any D-dimensional space and grid size N for different space-filling curves. A comparative study of different space-filling curves with respect to the description vector is conducted and results are presented and discussed.

1. INTRODUCTION

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 [15] where he introduces a mapping from the unit interval to the unit square. Hilbert [9] generalizes the idea to a mapping of the whole space. Following Peano and Hilbert curves, many space-filling curves are proposed, e.g., [4, 5]. For a historical survey and more types of space-filling curves, the reader is referred to [16].

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 [12]. The objective of this paper is to provide a systematic and a scalable framework for selecting the appropriate space-filling curve for any application. To achieve this objective, we divide any space-filling curve into segments. Each segment connects two consecutive multi-dimensional points. Thus, a D-dimensional space-filling curve with grid size N would have N^D-1 segments that connect N^D points. We distinguish among five different segment types *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still*. A space-filling curve SFC is described by its description vector $V = (J, C, R, F, S)$, where J, C, R, F, and S, are the percentages of *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still* segments, respectively. Then, with only looking at the description vector V, one can choose the right space-filing curve for a given application.

Although space-filling curves were discovered in the last century [9, 15], 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 in spatial join [14], range queries [6], nearest neighbor queries [11], spatial access methods [7], packing R-Tree [10], disk scheduling [1], and image processing [17]. The properties of different space-filling curves is explored in [2, 12, 13]. In [2], the properties of several space filling curves in the two- and three-dimensional spaces is discussed, and new measures to describe the behavior of any space-filling curve is presented. The notion of irregularity is presented in [12] as a quantitative measure of how irregular a space-filling curve is. In [13], the clustering properties of the Hilbert SFC is analyzed by deriving closed formulas for the number of clusters in a given query region.

The rest of this paper is organized as follows. Different types of segments in space-filling curves are presented in Section 2. Section 3 develops closed formulas to compute the description vector for the Peano, Gray and Hilbert SFCs. In Section 4, we conduct a comprehensive comparison among different space-filling curves. Finally, Section 5 concludes the paper.

*This work was supported in part by the National Science Foundation under Grants IIS-0093116, EIA-9972883, IIS-9974255, IIS-0209120, the NAVSEA/Naval Surface Warfare Center, Crane, and by Purdue Research Foundation.

2. SEGMENT TYPES IN SPACE-FILLING CURVES

A D-dimensional space-filling curve with grid size N has N^D-1 segments that connect N^D points. Each segment is classified as one or more of five segment types: *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still*. In this section, we give a precise definition of each segment type along with an iterative equation to compute the number of segments from each type for each dimension. For the rest of the paper, we use the term $P_i = (u_1, u_2, \cdots, u_k)$ to indicate the ith point in a space-filling curve. Also, $P_i.u_k$ indicates the kth dimension in the ith point in a space-filling curve.

2.1 Jump

Definition 1. A *Jump* in an SFC is said to happen when the distance, along any of the dimensions, between two consecutive points in the SFC is greater than one.

Formally, for any two consecutive multi-dimensional points P_i, P_{i+1} in an SFC, a *Jump* occurs in dimension k iff $abs(P_i.u_k - P_{i+1}.u_k) > 1$. The total number of *Jump* segments in a dimension k in a D-dimensional space with grid size N is: $J(k,N,D) = \sum_{i=0}^{N^D-2} f_J(i,k)$ where $f_J(i,k) = 1$ iff $abs(P_i.u_k - P_{i+1}.u_k) > 1$ and 0 otherwise. The total number of *Jump* segments in an SFC is: $J_T(N,D) = \sum_{k=0}^{D-1} J(k,N,D)$

A *Jump* in a space-filling curve reflects the locality of the consecutive points in the order implied by the space-filling curve. The lack of *Jump* segments indicates more ability for clustering. However, *Jump* may or may not be a favorable property based on the application type. For example, in a disk-head scheduling [1], *Jumps* are considered bad, as they result in a longer seek time without retrieving any data. On the other side, in multi-priority scheduling, *Jumps* are considered good, as the ability of fast moving among different priority types is required.

2.2 Contiguity

Definition 2. A *Contiguity* in an SFC is said to happen when the distance, along any of the dimensions, between two consecutive points in the SFC is equal to one.

Formally, for any two consecutive multi-dimensional points P_i, P_{i+1} in an SFC, a *Contiguity* occurs in dimension k iff $abs(P_i.u_k - P_{i+1}.u_k) = 1$. The total number of *Contiguity* segments in a dimension k in a D-dimensional space with grid size N is: $C(k,N,D) = \sum_{i=0}^{N^D-2} f_C(i,k)$ where $f_C(i,k) = 1$ iff $abs(P_i.u_k - P_{i+1}.u_k) = 1$ and 0 otherwise. The total number of *Contiguity* segments in an SFC is: $C_T(N,D) = \sum_{k=0}^{D-1} C(k,N,D)$

Contiguity reflects the ability of a space-filling curve to go continuously along any of the dimensions. A high ratio of *Contiguity* indicates a lower ratio in *Jump*. As in *Jumps*, *Contiguity* may or may not be favorable, depending on the underlying application.

2.3 Reverse

Definition 3. A segment in an SFC is termed a *Reverse* segment if the projection of its two consecutive points, along any of the dimensions, results in scanning the dimension in decreasing order.

Formally, for any two consecutive multi-dimensional points P_i, P_{i+1} in an SFC, a *Reverse* occurs in dimension k iff $P_{i+1}.u_k < P_i.u_k$. The total number of *Reverse* segments in a dimension k in a D-dimensional space with grid size N is: $R(k,N,D) = \sum_{i=0}^{N^D-2} f_R(i,k)$ where $f_R(i,k) = 1$ iff $P_{i+1}.u_k < P_i.u_k$ and 0 otherwise. The total number of *Reverse* segments in an SFC is: $R_T(N,D) = \sum_{k=0}^{D-1} R(k,N,D)$

A *Reverse* segment is also classified as either a *Jump* or a *Contiguity* one. Whether reverse segments is favorable or not relates to the semantic of the sorted parameter. For example, consider the real-time applications. When applying a space-filling curve to a deadline parameter, the sorting from the largest to the smallest, i.e., in reverse order, means that we visit the points with larger deadline before the points with smaller deadline. In this case, reverse ordering is considered unfavorable. As another example, consider the case of disk-head scheduling [1]. Based on the disk-head movement, alternating between forward and reverse ordering is favorable. In summary, it is important to point out and quantify whether or not a space-filling curve exhibits reverse ordering in its dimensions.

2.4 Forward

Definition 4. A segment in an SFC is termed a *Forward* segment if the projection of its two consecutive points, along any of the dimensions, results in scanning the dimension in increasing order.

Formally, for any two consecutive multi-dimensional points P_i, P_{i+1} in an SFC, a *Forward* occurs in dimension k iff $P_{i+1}.u_k > P_i.u_k$. The total number of *Forward* segments in a dimension k in a D-dimensional space with grid size N is: $F(k,N,D) = \sum_{i=0}^{N^D-2} f_F(i,k)$ where $f_F(i,k) = 1$ iff $P_{i+1}.u_k > P_i.u_k$ and 0 otherwise. The total number of *Forward* segments in an SFC is: $F_T(N,D) = \sum_{k=0}^{D-1} F(k,N,D)$

As in *Reverse* segment, a *Forward* segment is also classified as either a *Jump* or a *Contiguity* segment. A higher ratio of *Reverse* segments indicates a lower ratio of *Forward* segments.

2.5 Still

Definition 5. A segment in an SFC is termed a *Still* segment when the distance, along any of the dimensions, between the segment's two consecutive points in the SFC is equal to zero.

Formally, for any two consecutive multi-dimensional points P_i, P_{i+1} in an SFC, a *Still* occurs in dimension k iff $P_{i+1}.u_k = P_i.u_k$. The total number of *Still* segments in a dimension k in a D-dimensional space with grid size N is: $S(k,N,D) = \sum_{i=0}^{N^D-2} f_S(i,k)$ where $f_S(i,k) = 1$ iff $P_{i+1}.u_k = P_i.u_k$ and 0 otherwise. The total number of *Still* segments in an SFC is: $S_T(N,D) = \sum_{k=0}^{D-1} S(k,N,D)$

A segment is considered as a *Still* segment if it does not match any of the other types. *Still* segments is the closure of other types. For example, a segment that is neither a *Jump* nor a *Contiguity* is considered as a *Still*. Also, a segment that is neither a *Reverse* nor a *Forward* segment is considered as a *Still*. In general, the number of *Still* segments in a dimension k indicates the percent that this dimension is ignored to visit other dimensions. Unlike other segment types, a *Still* segment can not be classified as another segment type.

2.6 Relation between segment types

The five segment types can be divided into two categories. The first category, termed the *distance category*, is concerned with the segment length. This includes *Jump*, *Contiguity*, and *Still* segments where the segment length is greater than, equal, or less than one, respectively. The second category, termed the *direction category* is concerned with the direction of the segment. This includes *Reverse*, *Forward*, and *Still* segments. Notice that the *Still* segments belong to the two categories where it serves as the closure of each category. The relation between segment types is summarized in the following Lemma.

LEMMA 1. *For any dimension k in a D-dimensional space with grid size N, the following equalities always hold.*

$$J(k,N,D)+C(k,N,D)+S(k,N,D)=N^D-1$$
$$R(k,N,D)+F(k,N,D)+S(k,N,D)=N^D-1$$
$$J_T+C_T+S_T=D(N^D-1)$$
$$R_T+F_T+S_T=D(N^D-1)$$

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

From Lemma 1, we can derive the following corollary.

COROLLARY 1. *To compute the description vector V, it is enough to compute only three segment types with at least one from each category. The other two segment types can be computed from Lemma 1*

3. CASE STUDIES

The time complexity for calculating the number of segments of any type in a D-dimensional space with grid size N is $O(N^D)$. Consider the case of 20 dimensions with grid size 16, we need 16^{20} operations to compute the number of *Jumps* of a space-filling curve. To avoid this excessive operation, we derive closed formulas that compute the number of segments of each type for any dimension k in a D-dimensional space with grid size N. In this paper, we concentrate only on the following 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 [3]. For the rest of the paper we use the term $V_T=(J_T,C_T,R_T,F_T,S_T)$ to denote the *total description vector*, where J_T,C_T,R_T,F_T, and S_T represent the total number of *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still* segments, respectively. Notice that $V=V_T/D(N^D-1)$.

3.1 Case Study I: The Peano SFC

The Peano SFC is introduced by Peano [15] and is also known as Morton encoding, quad code, bit-interleaving, N-order, locational code, or Z-order. The Peano SFC is constructed recursively as in Figure 1. The basic step (Figure 1a) contains four points in the four quadrants of the space. Figure 1b contains four repeated blocks of Figure 1a at a finer resolution and is visited in the same order as in Figure 1a. Similarly, Figure 1c contains four repeated blocks of Figure 1b at a finer resolution. For details about extending the Peano SFC to multi-dimensional space, the reader is referred to [12].

LEMMA 2. *In a D-dimensional space with grid size N, the number of Jump, Contiguity, Reverse, Forward, and Still segments in any dimension k for the Peano SFC is:*

$$J(k,N,D)=\frac{\left(N^D-2^{2D}\right)\left(2^D-2\right)}{2^{2D-k}\left(2^D-1\right)}+2^k-1$$
$$S(k,N,D)=N^D\left(1-2^{k-D+1}\right)$$
$$C(k,N,D)=N^D-1-J(k,N,D)-S(k,N,D)$$
$$R(k,N,D)=\frac{2^k\left(N^D-2^D\right)\left(2^D-2\right)}{2^D\left(2^D-1\right)}+2^k-1$$
$$F(k,N,D)=N^D-1-R(k,N,D)-S(k,N,D)$$

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

LEMMA 3. *The total description vector V_T for the D-dimensional Peano SFC with grid size N is $V_T=(J_T,C_T,R_T,F_T,S_T)$ where:*

$$J_T=\left(\frac{N}{2}\right)^D\left(1-2^{1-D}\right)+1-D$$
$$C_T=\left(\frac{N}{2}\right)^D\left(2^{D+1}+2^{1-D}-3\right)$$
$$R_T=N^D\left(1-2^{1-D}\right)+1-D$$
$$F_T=N^D-1$$
$$S_T=N^D\left(2^{1-D}+D-2\right)$$

The description vector $V=V_T/D(N^D-1)$

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

3.2 Case Study II: The Gray SFC

The Gray SFC uses the Gray code representation in contrast to the binary code representation as in the Peano SFC. Figure 1 gives the recursive construction of the Gray SFC. The basic step (Figure 1d) contains four points in the four quadrants of the space. Figure 1e contains four repeated blocks of Figure 1d 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 1d, while the second and the third blocks are constructed by rotating the block of Figure 1d by 180^0. Similarly, Figure 1f is constructed from two blocks of Figure 1e at a finer resolution and two blocks of the rotation of Figure 1e by 180^0. For details about extending the Gray SFC to multi-dimensional space, the reader is referred to [12].

LEMMA 4. *In a D-dimensional space with grid size N, the number of Jump, Contiguity, Reverse, Forward, and Still segments in any dimension k for the Gray SFC is:*

$$J(k,N,D)=\frac{\left(N^D-2^D\right)}{2^{D-k}\left(2^D-1\right)},\qquad C(k,N,D)=\frac{N^D}{2^{D-k}}$$
$$S(k,N,D)=N^D-1-J(k,N,D)-C(k,N,D)$$
$$R(0,N,D)=\frac{N^D-2^D}{2\left(2^D-1\right)}$$
$$R(k,N,D)=\frac{2^{k-1}\left(N^D-1\right)}{2^D-1},\qquad k>0$$
$$F(k,N,D)=N^D-1-R(k,N,D)-S(k,N,D)$$

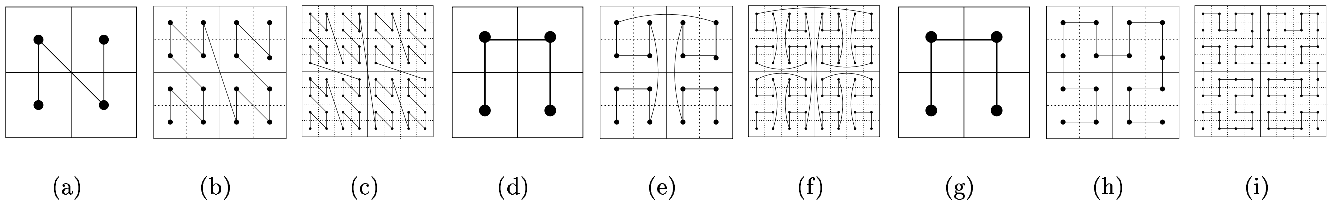


Figure 1: Recursive SFCs: (a),(b),(c) The Peano SFC. (d),(e),(f) The Gray SFC. (g),(h),(i) The Hilbert SFC.

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

LEMMA 5. *The total description vector V_T for the D-dimensional Gray SFC with grid size N is $V_T = (J_T, C_T, R_T, F_T, S_T)$ where:*

$$J_T = \left(\frac{N}{2}\right)^D - 1 \qquad C_T = \left(\frac{N}{2}\right)^D \left(2^D - 1\right)$$

$$R_T = \frac{N^D - 2}{2} \qquad F_T = \frac{N^D}{2}$$

$$S_T = (D-1)(N^D - 1)$$

The description vector $V = V_T / D(N^D - 1)$

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

3.3 Case Study III: The Hilbert SFC

Figure 1 gives the recursive construction of the Hilbert SFC. The basic block of the Hilbert SFC (Figure 1g) is the same as the basic block of the Gray SFC (Figure 1d). The basic block is repeated four times at a finer resolution in the four quadrants, as given in Figure 1h. The quadrants are visited in their gray order. The second and third blocks in Figure 1h have the same orientation as in Figure 1g. The first block is constructed from rotating the block of Figure 1g by 90^0, while the fourth block is constructed by rotating the block of Figure 1g by -90^0. Figure 1i is constructed from Figure 1h in an analogous manner.

LEMMA 6. *In a D-dimensional space with grid size N, the number of Jump, Contiguity, Reverse, Forward, and Still segments in any dimension k for the Hilbert SFC is:*

$$J(k, N, D) = 0$$

$$C(k, N, D) = \sum_{i=1}^{D-1} C((k+i) mod D, \frac{N}{2}, D) + 2C(k, \frac{N}{2}, D) + 2^k$$

$$C(k, 1, D) = 0$$

$$S(k, N, D) = N^D - 1 - J(k, N, D) - C(k, N, D)$$

$$R(0, N, D) = (C(0, N, D) - N + 1)/2$$

$$R(k, N, D) = C(k, N, D)/2, \qquad k > 0$$

$$F(k, N, D) = N^D - 1 - R(k, N, D) - S(k, N, D)$$

PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

LEMMA 7. *The total description vector V_T for the D-dimensional Hilbert SFC with grid size N is $V_T = (J_T, C_T, R_T, F_T, S_T)$ where:*

$$J_T = 0 \qquad C_T = N^D - 1$$

$$R_T = \frac{N}{2}\left(N^{D-1} - 1\right) \qquad F_T = \frac{N}{2}\left(N^{D-1} + 1\right) - 1$$

$$S_T = (D-1)(N^D - 1)$$

The description vector $V = V_T / D(N^D - 1)$

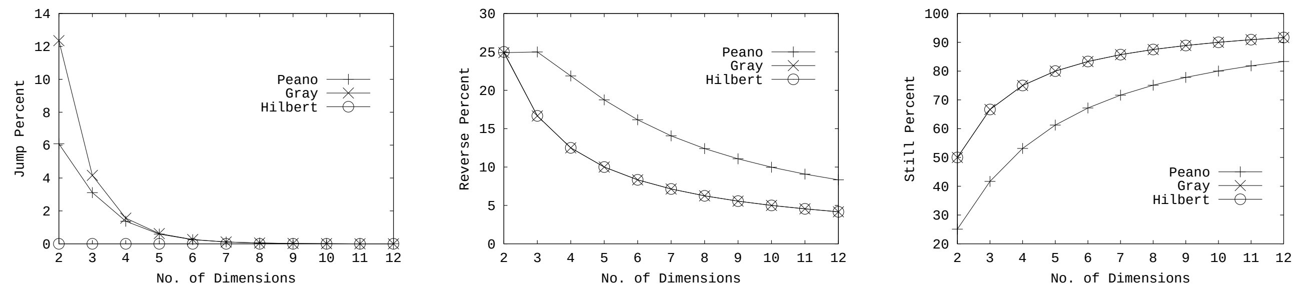
PROOF. The proof is omitted for brevity. The reader is referred to [3] for a detailed proof. □

4. PERFORMANCE EVALUATION

In this section, we perform comprehensive experiments to compare the Peano, Gray, and Hilbert SFCs with respect to different segment types. The results in this section are computed using the closed formulas developed in Section 3. Notice that it is timely infeasible to compute segment types in high-dimensional space using the definition and iterative equations from Section 2. For more experiments and comparison, the reader is referred to [3].

4.1 Scalability of Space-Filling Curves

In this section, we address the issue of scalability, e.g., when the number of dimensions and/or the number of points per dimension increases. For the following experiments, we use Lemmas 3, 5, and 7 to compute the description vector V. Figure 2 gives the result of setting the grid size N=16, while measuring different segment types (*Jump*, *Reverse*, and *Still*) for up to 12 dimensions. An interesting result appears in the *Jump* segments (Figure 2a) where both the Peano and Gray SFCs have very low percentage (almost 0%) of *Jumps* after six dimensions while the Hilbert SFC has no *Jumps* for any dimensions. The fact that the Hilbert SFC has no *Jumps* is well-known [8, 13], and it is the main criteria for choosing the Hilbert SFC in many applications e.g., [8, 10]. However, this experiment emphasizes that both the Peano and Gray SFCs share the property of no *Jumps* with the Hilbert SFC for medium and high dimensionality. The trend of *Contiguity* and *Forward* segments is similar to the *Jump* and *Reverse* segments, respectively, although they are different in the percent value. The Gray and Hilbert SFCs have similar behavior for all segment types except for low-dimensionality in the *Jump* and *Contiguity* segments. Notice that all segment types except *Still* are decreasing as the number of dimensions increases. The reason for this comes from the *Still* definition. A *Still* segment indicates that the value of one of its dimensions does not change. With a larger number of dimensions, it is difficult to find a segment that connects two consecutive multi-dimensional points that are



(a) Jump (b) Reverse (c) Still

Figure 2: Scalability of space-filling curve w.r.t dimensionality.

different in all dimensions. Thus, almost each segment is counted as a *Still* for one or more dimensions.

The three space-filling curves almost have constant performance regardless of grid size. This can be noted from the description vector V, where getting the $\lim_{N\to\infty} V$ gives a constant value that does not depend on N.

4.2 Fairness of Space-Filling Curves

In this section, we test the fairness[1] of space-filling curves. For each segment type T, we use the standard deviation of the number of T segments over all dimensions as an indication for fairness. The lower the standard deviation the more fair the space-filling curve is. For the experiments of this section, we use Lemmas 2, 4, and 6 to compute the number of segments of each type for each individual dimension rather than the total that is used in the description vector.

Figure 3 gives the standard deviation of *Jump*, *Contiguity*, and *Forward* segments. *Still* and *Reverse* segments have the same figure as in *Contiguity*. It is clear that for all segment types, the Hilbert SFC is the most fair space-filling curve with very low standard deviation. Also, except for the *Jump* segments, the Peano SFC gives the worst behavior. The interesting result is that both the Peano and Gray SFCs tend to be more fair as the dimensionality increases while the Hilbert SFC do the opposite. This indicates that for very high dimensionality, the Hilbert SFC may not be the most fair space-filling curve.

4.3 Intentional Bias of Space-Filling Curves

A very 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[2] of each space-filling curve by plotting its behavior for each dimension individually. Figure 4 gives the intentional bias for direction segments. The experiment is performed for four-dimensional space with grid size 16. Each dimension is plotted individually as a stacked bar that contains the percent of distance or direction segments. The fifth column is the percent of the total number of segments over all dimensions from each type. Note that the height of each bar is 100 (refer to Lemma 1).

From Figure 4c, the Hilbert SFC is not biased to any dimension. The number of *Reverse*, *Forward*, and *Still* segments is almost equal for all dimensions. This agrees with the results from the previous section, where the Hilbert SFC has a very low standard deviation. With respect to *Reverse* and *Forward*, the Peano SFC is biased toward the last dimension where the number of these segments is increased with the dimension number k. On the other side, the number of *Still* segment decreases with the dimension number. The Gray SFC has similar behavior as in the Peano SFC, however, the increase/decrease in *Reverse* and *Forward*/ *Still* segments is slower. The same analysis applies to distance segments. The Hilbert SFC is extremely fair, while in the Peano and Gray SFCs, the number of *Contiguity* segments is increasing rapidly with the dimension number. The number of *Jump* segments is very low compared to *Contiguity* and *Still* segments.

5. CONCLUSION

Space-filling curves are used as a mapping scheme from the multi-dimensional space into the one-dimensional space. The behavior of different space-filling curves in the D-dimensional space is analyzed. A description vector V is proposed to give a brief description for each space-filling curve. Closed formulas that depend on the space dimensionality and grid size are derived to compute V. The main idea is to divide the space-filling curve into a set of connected segments. Each segment connects two consecutive multi-dimensional points. Five segment types are distinguished, namely, *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still*. The description vector V contains the percent of each segment type. Several experiments are conducted to show the scalability and fairness of space-filling curves with respect to segment types.

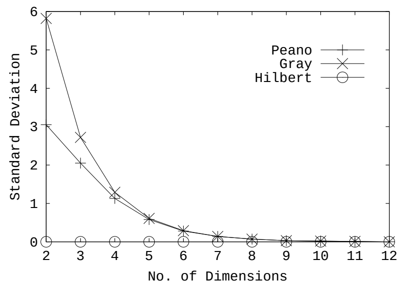
6. REFERENCES

[1] W. G. Aref, K. El-Bassyouni, I. Kamel, and M. F. Mokbel. Scalable qos-aware disk-scheduling. In *International Database Engineering and Applications Symposium, IDEAS*, Alberta, Canada, July 2002.

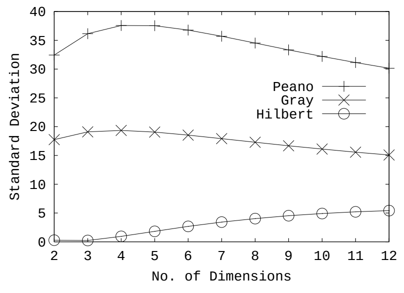
[2] W. G. Aref and I. Kamel. On multi-dimensional sorting orders. In *Proc. of the 11th Intl. Conf. on Database and Expert Systems Applications, DEXA*, pages 774–783, London, Sept. 2000.

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

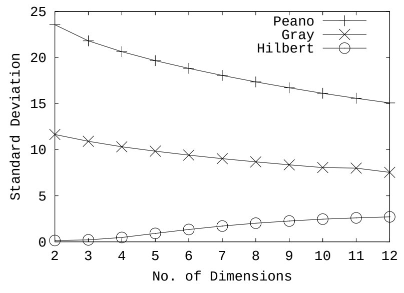
[2]We say that a space-filling curve is intentionally biased to a dimension k with respect to segment type T if it has more T segments in dimension k with respect to all other dimensions.



(a) Jump

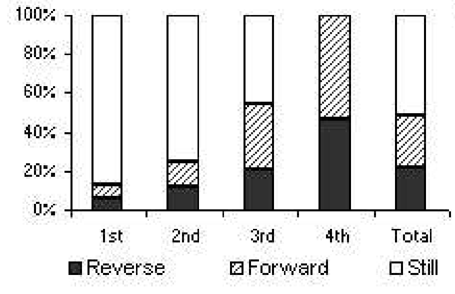


(b) Contiguity

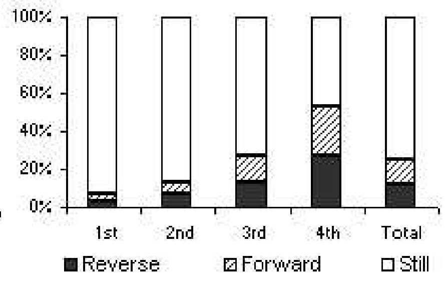


(c) Forward

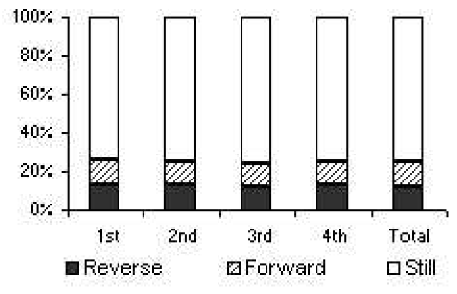
Figure 3: Fairness of space-filling curves.



(a) Peano



(b) Gray



(c) Hilbert

Figure 4: Intentional bias of space-filling curves w.r.t direction segments.

[3] 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.

[4] 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.

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

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

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

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

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

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

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

[12] M. F. Mokbel and W. G. Aref. Irregularity in multi-dimensional space-filling curves with applications in multimedia databases. In *Proc. of the 2nd Intl. Conf. on Information and knowledge Management, CIKM*, pages 512–519, Atlanta, GA, Nov. 2001.

[13] 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.

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

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

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

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