# Optimality of Clustering Properties of Space Filling Curves

PAN XU, Dept. of Computer Science, University of Maryland, College Park
SRIKANTA TIRTHAPURA, Dept. of Electrical and Computer Engg., Iowa State University

Space filling curves have been used in the design of data structures for multidimensional data since many decades. A fundamental quality metric of a space filling curve is its "clustering number" with respect to a class of queries, which is the average number of contiguous segments on the space filling curve that a query region can be partitioned into. We present a characterization of the clustering number of a general class of space filling curves, as well as the first non-trivial lower bounds on the clustering number for any space filling curve. Our results answer questions that have been open for more than 15 years.

Categories and Subject Descriptors: H.3.1 [**Content Analysis and Indexing**]: Indexing Methods

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: clustering, Hilbert curve, lower bound, space filling curves, Z curve

## 1. INTRODUCTION

Many query processing techniques for multidimensional data are based on a *space filling curve* (SFC), which is a bijection from points in a discrete multidimensional universe to a one dimensional universe of the same cardinality. For example, Orenstein and Merrett [Orenstein and Merrett 1984] proposed the use of SFCs for answering range queries on multidimensional data: *Preprocess (index) a set of input points $P$ such that when presented with a query box $Q$, it is possible to quickly compute a function of the set of all points in $P$ that fall in $Q$*. The advantage of an SFC is that conventional data structures that were used to organize one dimensional data can be directly used on higher dimensional data. Gutman [Gutman 1999] in his survey on space-filling curves in geospatial applications points to the advantage that SFCs can give good performance for geospatial applications without requiring any modification to the underlying DBMS code. The simplicity and elegance of this idea has caused it to become very popular, and now there are numerous databases that use SFCs to organize multidimensional data, including Oracle Spatial [Oracle ].

When data is ordered according to an SFC, a query region in multidimensional space will be partitioned into some number of segments on the SFC, and all such segments need to be retrieved and examined in order to process the query. It is desirable that a query region be partitioned into a small number of "clusters" such that each cluster consists of points that are contiguously ordered by the SFC. This leads us to define the "clustering number" of an SFC $\pi$ with respect to a given query region $q$ as the *smallest number of clusters into which $q$ can be partitioned such that the points within a cluster are ordered consecutively by the SFC.* When processing a query on data stored on the disk, the clustering number is a measure of the number of disk "seeks" that need to be

This work is partially supported by the National Science Foundation, under grants 0834743, 0831903. This work was performed when Pan Xu was with the Dept. of Electrical and Computer Engg. at Iowa State University.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

performed in order to process the query. Since a disk seek is an expensive operation, this is a significant and useful metric to have. The smaller the clustering number of a query, the better is the performance of the index. Note that in this work, we are concerned with a *discrete* multidimensional universe, and the space filling curves that we consider are also discrete entities which impose an ordering among all cells in the universe.

In an influential work, Moon *et al.* [Moon et al. 2001] presented an analysis of the clustering number of the Hilbert SFC. They showed that the average number of clusters on the Hilbert curve due to a "rectilinear polyhedron" query was equal to the surface area of the polyhedron divided by two times the number of dimensions. Since the publication of this work, it has received more than 300 citations. But even after a decade since this work, and more than two decades of interest in the clustering number of an SFC, many basic questions about clustering on SFCs remain unanswered. In particular:

(1) **General Techniques:** Are there any general methods for analyzing the clustering number of an SFC? What is the performance of commonly used SFCs, such as the $Z$-curve, over a given class of queries?
(2) **Lower Bound:** Are there lower bounds on the clustering number of an SFC? This question has been raised before by Jagadish [Jagadish 1997] in the context of a $2 \times 2$ square query region, but no non-trivial lower bounds were known so far.
(3) **Optimality:** It is a widely held belief that the Hilbert curve achieves the best possible clustering, on average. For what classes of queries is the Hilbert curve optimal? For what classes of queries is it sub-optimal?

### 1.1. Contributions

In this work, we present substantial progress towards answering the above questions. We consider two basic query types, a multidimensional *rectangular* query, formed by the intersection of half planes, and a *rectilinear* query, which is formed by the union of multidimensional rectangles. For both query classes, it assumed that the size of the bounding box of the query (the smallest rectangle that contains the query) is a constant that does not grow with the size of the universe. We consider the average clustering number on a set of queries formed by applying all possible *translations* and one or more *rotations* on a single query. Note that we only consider rotations by 90 degrees along different combinations of dimensions – a precise definition of rotations is presented in Section 2.

*General Method.* We present a general method that helps in analyzing the average clustering number. The main idea is as follows. An SFC can be viewed as a path, i.e. a set of edges, that traverses all cells of the universe. Instead of considering a query centric view, where we compute the clustering number of individual queries and average over all of them, we use an *edge centric* view, where we compute the contribution of each edge in the SFC to the clustering numbers of all queries. This edge centric view substantially simplifies the analysis, and leads to the following results.

*Lower Bound.* We present a lower bound on the clustering number of *any* SFC for the class of rectangular queries, for any set of rotations (Theorem 5.1). This answers a more general version of the question raised by Jagadish [Jagadish 1997]. Prior to our work, only upper bounds on the clustering number of specific SFCs, such as the Hilbert SFC were known. Further, we show that for the set of rectangular queries, for any subset of rotations, there is always a continuous SFC that achieves optimal clustering (Theorem 5.2).

*Optimal SFC for Rectangular Queries.* We present Algorithm 1 which can generate an optimal SFC given as an input a set of rectangular queries formed by all possible translations and any subset of rotations. In a practical setting, this is useful in designing an index for range queries for which the expected query mix is known in advance.

*Exact Characterization of Continuous SFCs.* We consider a class of SFCs that we call "continuous SFCs", which have the property that neighbors along the SFC are also nearest neighbors in the high-dimensional grid. For any rectilinear query $g$, and any set of rotations, we show that the clustering number of any SFC can be expressed as a simple formula involving the scalar product of two vectors, one derived from the query shape and the set of rotations, and the other derived from the space filling curve itself (Theorems 4.2, 4.4).

When all possible rotations are considered, we show that *every continuous SFC is optimal for rectangular queries* (Theorem 4.8). The result of Moon *et al.* [Moon et al. 2001] on the analysis of the Hilbert curve follows as a special case of our result on continuous SFCs.

*Analysis of the $Z$ curve.* We then consider the popular $Z$ curve. Our earlier analysis of continuous SFCs does not apply here since the $Z$ curve is not continuous. We derive an exact expression for the performance of the $Z$ curve in Theorem 6.1. We find that the performance of the $Z$ curve for cube shaped queries is nearly a factor of $2$ worse than the optimal.

*Non-Continuous SFCs and Non-Rectangular Queries.* We then investigate properties of SFCs that are not necessarily continuous. We define a class of SFCs called "near-continuous SFCs", which are a slight generalization of continuous SFCs. We show that for certain natural queries that we call "connected queries", near-continuous SFCs can outperform the best continuous SFC. Further, for certain natural connected queries, the best near-continuous SFC can be outperformed by an SFC that is not near-continuous (see Sections 7.1, 7.2).

The above shows that for the case of queries that are not rectangular, there may not be an "easily defined" class of SFCs that will contain the optimal SFC. This is to be contrasted with the case of rectangular queries, for which there always exists a continuous SFC that is optimal.

### 1.2. Related Work

Moon *et al.* [Moon et al. 2001] considered an analysis of the Hilbert curve in $d$ dimensions. Similar to our model, they also considered the query class of all translations of any rectilinear query $g$, and showed the elegant result that as $n$, the size of the universe, approaches $\infty$, the clustering number of the Hilbert curve approaches the surface area of the query $g$, divided by twice the number of dimensions. Since the Hilbert curve is a continuous curve, our analysis of a continuous curve applies here. In particular, Corollary 4.9 implies the result of [Moon et al. 2001].

Bugnion *et al.* [Bugnion et al. 1997] present an analysis of the clustering number of space filling curves in two dimensions. For a class of curves they define as "continuous", they provide a formula for computing the clustering number for rectangular queries. We note that according to their definition of a continuous SFC, the curve is allowed to move from one cell to another cell whose coordinates differ by no more than one along any dimension, while in our definition, the curve is allowed to move from one cell to another cell whose coordinate differs by one along exactly one dimension. Their definition of continuous SFCs corresponds to our definition of "near-continuous" SFCs (see Section 7.1). Our analysis in this work generalizes that of [Bugnion et al. 1997] in the following respects. We consider SFCs in $d$ dimensions for constant $d$, while their analysis is for two dimensions only. Next, we consider the analysis of rectilinear queries, as well as their subclass of connected queries, while their analysis is restricted to rectangular queries. Further, we present a lower bound on the clustering number for any arbitrary SFC (whether continuous or not), while their analysis is only restricted to upper bounds, and they do not consider SFCs that are not continuous.

Xu and Tirthapura [Xu and Tirthapura 2012] present an analysis of clustering properties of space filling curves. This work enhances [Xu and Tirthapura 2012] in the following ways:

- We present an analysis of the clustering properties of the popular $d$-dimensional $Z$ SFC. Our general results on continuous SFCs do not hold for the $Z$ SFC, and hence we conduct this analysis from first principles, through a detailed consideration of the structure of the $Z$ curve; this also shows the power of our basic techniques in analyzing different types of SFCs, not just continuous ones.
- We consider the clustering properties of near-continuous SFCs and presents a result on their performance (Theorem 4.4), while prior work did not consider this class of SFCs.
- We consider a class of queries called *connected queries*, which are broader than the class of rectangular queries, yet not as general as rectilinear queries. For such queries, we show that there exist cases when near-continuous SFCs can do significantly better than continuous SFCs, and further, there exist cases when SFCs that are not near-continuous can significantly outperform near-continuous SFCs.

Jagadish [Jagadish 1997] considered the clustering performance of the two-dimensional Hilbert curve on a $\sqrt{n} \times \sqrt{n}$ universe when the query region was a $m \times m$ square. For $2 \times 2$ queries, he derived that the average clustering number approaches $2$ as $n$ approaches $\infty$. He says "We conjecture that this number 2 is an asymptotic optimum. $\cdots$. Proving this conjecture is a subject for future research". Our results show that the optimum clustering number for a $2 \times 2$ square over any SFC is indeed equal to $2$. Our analysis considers lower bounds for a more general problem, where the SFC is over a general multidimensional universe, and the query is any rectangle.

Asano *et al.* [Asano et al. 1997] present an analysis of the clustering properties of SFCs in two dimensions in a model that is different from ours, as follows. For a query $q$ consisting of $|q|$ cells, the query processor is allowed to return a set of $C|q|$ cells which is a *superset* of $q$, and can be divided into a small number of clusters, where $C$ is a constant greater than $1$. In contrast, in our model, we require the query processor to return the set of exactly the cells present in the query $q$, and consider the number of clusters thus created. A similar approach is taken by Haverkort [Haverkort 2011], who considers a model where, given a query region $q$ (a disk, in his case), a small set of clusters (referred to as "tiles" in the paper) is found whose union is a superset of $q$, but not much larger than $q$. In our model the number of clusters is always greater than or equal to the number of clusters in the model of Asano *et al.* and Haverkort.

There is a large literature on SFCs that we will not attempt to cite here, but to our knowledge, no previous work has considered lower bounds and a general analysis of clustering properties of SFC as we do here. Alber and Niedermeier [Alber and Niedermeier 2000] present a precise characterization of Hilbert curves in dimensions $d \geq 3$. Many other applications of SFCs to problems on spatial data can be found in literature, including databases [Jagadish 1990], data partitioning in scientific computing [Warren and Salmon 1993; Pilkington and Baden 1996], parallel domain decomposition [Aluru and Sevilgen 1997; Pilkington and Baden 1996], and cryptography [Matias and Shamir 1987].

**Organization of the Paper:** The rest of this paper is organized as follows. We define the model and the problem in Section 2. In Section 3, we present a general technique for computing the clustering number of an SFC for any class of queries, which forms the basis for further analysis. We present the results for a continuous SFC in Section 4, and the lower bound on any SFC in Section 5. In Section 6 we present an analysis of the $Z$ curve. In Section 7, we present results for near-continuous SFCs (Section 7.1), results for connected queries (Section 7.2), and clarifications on the query models (Section 7.3).

## 2. MODEL AND PRELIMINARIES

Let $U$ denote the $d$ dimensional $\sqrt[d]{n} \times \cdots \times \sqrt[d]{n}$ grid of $n$ cells. Each point in $U$ is a $d$-tuple $(x_1, x_2, \ldots, x_d)$ where for each $i = 1 \ldots d$, $0 \leq x_i < \sqrt[d]{n}$. For $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$, the Manhattan distance between them is defined to be $\sum_{i=1}^{d} |x_i - y_i|$. A given SFC may impose special requirements on the grid size. For example, the Hilbert curve and the $Z$ curve are defined for a grid whose side length is a power of $2$ while the Peano curve expects the grid size to be a power of $3$. Our analysis does not place any restriction on the grid size of the universe, though in dealing with specific SFCs we assume that the grid size is consistent with the definition of the SFC being considered.

*Definition* 2.1. An SFC $\pi$ on $U$ is a bijective mapping $\pi : U \to \{0, 1, \cdots, n-1\}$.

Some popularly used space filling curves are the Z-curve [Orenstein and Merrett 1984; Morton 1966] (also known as the Morton ordering), the Hilbert curve [Hilbert 1891], and the Gray code curve [Faloutsos 1986; 1988]. Figures 1(a), 1(b), 1(c), and 1(d) show the Hilbert curve, the row-major curve, the $Z$ curve, and the Peano curve respectively.

*Definition* 2.2. An SFC $\pi$ is said to be a continuous SFC if it has the property that for every $0 \leq i \leq n-2$, the Manhattan distance between $\pi^{-1}(i)$ and $\pi^{-1}(i+1)$ is $1$.

In other words, a continuous SFC always travels from one cell on the grid to another cell that is at a Manhattan distance of 1. According to Definition 2.2, the row-major curve (Figure 1(b)) and the Hilbert curve are both continuous SFCs while the Z-curve is not.

For any SFC $\pi$, and dimension $i$, $1 \leq i \leq d$, let $N^i(\pi)$ be the set of pairs $(\alpha, \beta) \in U \times U$ such that (1) $\pi(\beta) = \pi(\alpha) + 1$, and (2) $(\alpha, \beta) \in E_i(U)$. Informally, the set $N^i(\pi)$ is the set of all edges of $\pi$ that connect points in $U$ that are at a Manhattan distance of $1$, and lie along dimension $i$.

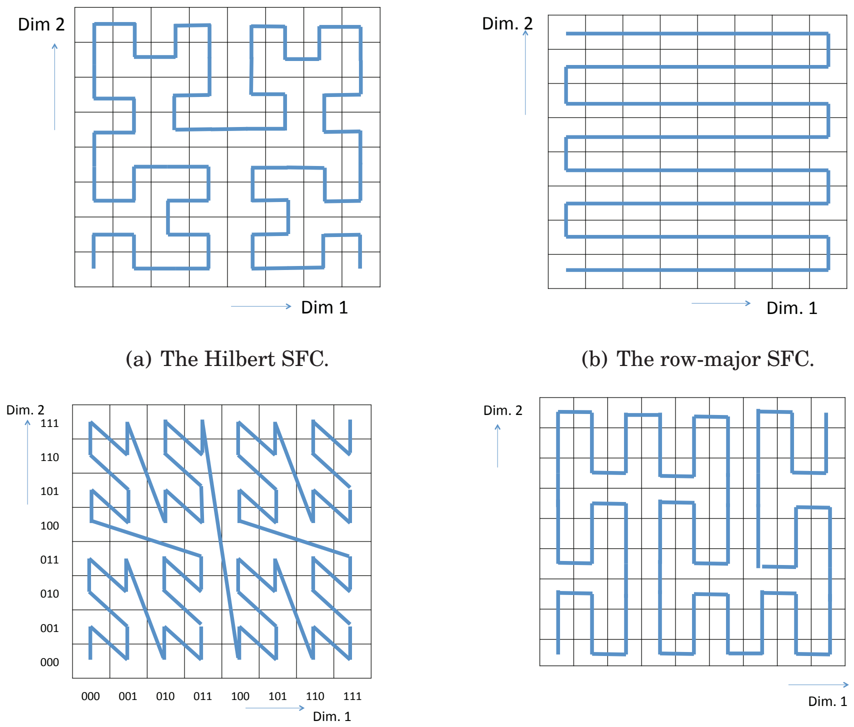
*Definition* 2.3. For an SFC $\pi$, vector $\mu(\pi)$ of length $d$ is defined as: $\mu(\pi) = (\mu_1(\pi), \mu_2(\pi), \ldots, \mu_d(\pi))$, where for $i = 1 \ldots d$

$$\mu_i(\pi) = \lim_{n \to \infty} \frac{|N^i(\pi)|}{n-1}$$

For many popular recursively-defined SFCs, vector $\mu(\pi)$ exists. For example, consider the two-dimensional Hilbert curve $\mathcal{H}$, row-major curve $R$, $Z$ curve and Peano curve $Pe$ which are shown in Figures 1(a), 1(b), 1(c), and 1(d), respectively.

We have $\mu_1(\mathcal{H}) = \mu_2(\mathcal{H}) = 1/2, \mu_1(R) = 1, \mu_2(R) = 0, \mu_1(Z) = 0, \mu_2(Z) = 1/2, \mu_1(Pe) = 2/9, \mu_2(Pe) = 7/9$. In this paper, we assume the vector $\mu(\pi)$ exists for all the SFCs that we consider. This assumption will allow a compact statement of the results on the clustering numbers of SFCs.

*Definition* 2.4. A set of cells $C \subseteq U$ is said to be a "cluster" of an SFC $\pi$ if the cells of $C$ are numbered consecutively by $\pi$.



(a) The Hilbert SFC.

(b) The row-major SFC.

(c) The $Z$-SFC.

(d) The Peano SFC.

Fig. 1. Some popular SFCs in two dimensions.

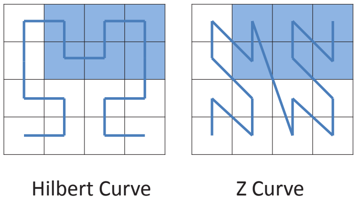


Fig. 2. For the same query region shown, the Hilbert curve has a clustering number of 1, and the $Z$ curve has a clustering number of 2.

For instance, the universe $U$ is a cluster for any SFC.

**Queries:** A query $q$ is any subset of $U$. The volume of query $q$, denoted $|q|$, is the number of cells in it. A rectangular query is a set of cells of the form $\{(x_1, x_2, \ldots, x_d) | \ell_i \leq x_i \leq \hbar_i, \quad \text{for each} \quad i = 1 \ldots d\}$. For a query $q$ which may not be a rectangle, the *bounding box* of $q$ denoted $B(q)$, is the smallest rectangle that contains all cells in $q$. In particular, if $q$ is a rectangle then $B(q)$ is equal to $q$. We say that a query $g \subset U$ is of a *fixed size* if the volume of $B(g)$ is independent of $n$, the size of the universe.

*Definition* 2.5. The clustering number of an SFC $\pi$ for a query $q$, denoted $c(q, \pi)$, is defined as the minimum number of clusters of $\pi$ that $q$ can be partitioned into.

See Figure 2 for an illustration of the above definition. It is not very interesting to consider the clustering number of an SFC with respect to a single query, for the following reasons. First, it is rarely the case that there is only one query of interest we need to optimize for. Second, it is easy to come up with an SFC that yields the optimal clustering (one cluster) for a specific query. Thus, we always consider the average clustering number of an SFC with respect to a set of queries.

*Definition* 2.6. The average clustering number of an SFC $\pi$ for a non-empty set of queries $Q$, denoted by $c(Q, \pi)$, is defined as:

$$c(Q,\pi) = \frac{\sum_{q \in Q} c(q,\pi)}{|Q|}$$

**Query Sets.** The set of queries that we consider is constructed as follows. We first consider a basic query, for example, a two dimensional rectangle $r$ consisting of the cells $\{(2+i, 3+j) | 1 \leq i \leq 2, 1 \leq j \leq 3\}$. Then we consider all possible *translations* of this shape $r$ combined with a set of one or more *rotations* along the different axes, to arrive at a set of queries.

We handle rotation by treating it as a permutation of the coordinates along different dimensions. For example, in two dimensions, a $2 \times 3$ rectangle can be rotated to a $3 \times 2$ rectangle by interchanging dimensions 1 and 2. More precisely, let $\Lambda^*$ be the set of all possible permutations of $(1, 2, \cdots, d)$. For $\lambda \in \Lambda^*$, and $1 \leq i \leq d$, let $\lambda(i)$ denote the image of $i$ under $\lambda$. For any $\lambda \in \Lambda^*$, we define the rotation of a cell $x \in U$ under $\lambda$ as: $\mathcal{P}(x = (x_1, \ldots, x_d), \lambda) = (x_{\lambda(1)}, x_{\lambda(2)}, \ldots, x_{\lambda(d)})$ The rotation of any query $g \subseteq U$ with $\lambda$ is defined as: $\mathcal{P}(g, \lambda) = \{\mathcal{P}(v, \lambda) | v \in g\}$

In some cases, permutation of coordinates can lead to a rotation along with a reflection along certain axes. While other definitions of rotations are also possible, we consider this definition for its simplicity. Similar results will hold for other definitions, since our analysis in essence holds for any set of queries all of which have the same number of cells.

For a query $g \subseteq U$, given a $d$ dimensional vector $\delta = (\delta_1, \delta_2, \ldots, \delta_d)$, the translation of $g$ subject to $\delta$ yields a new query equal to $\{v + \delta | v \in g\}$ (note that "+" denotes vector addition). For query $g$, the set of all possible translations of $g$, denoted by $\mathcal{T}(g)$, is the set of all possible queries that can be obtained by a translation of $g$ (see Figure 3):

*Definition* 2.7. For a query $g$ and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$, the query set $\mathcal{Q}(g, \Lambda)$ is defined as

$$\mathcal{Q}(g,\Lambda) = \bigcup_{\lambda \in \Lambda} \mathcal{T}(\mathcal{P}(g,\lambda))$$

For simplicity, we interpret the above to be a multiset, where we consider the queries in $\mathcal{T}(\mathcal{P}(g, \lambda_1))$ and $\mathcal{T}(\mathcal{P}(g, \lambda_2))$ to be distinct as long as $\lambda_1 \neq \lambda_2$. Note that it is possible that $\lambda_1 \neq \lambda_2$, but $\mathcal{T}(\mathcal{P}(g, \lambda_1))$ and $\mathcal{T}(\mathcal{P}(g, \lambda_2))$ are the same set of queries. For example, $g$ may be a single cell whose coordinates are the same along all dimensions, so that any rotation $\lambda$ makes no difference. Our results for rectangular queries can be extended in a straightforward manner to the case when we do not consider the multiset union above, but a regular set union, as we detail in Section 7.3.

For example, suppose $d = 2$ and $r$ is a $2 \times 3$ rectangle, with length 2 along dimension 1 and 3 along dimension 2. Then $\Lambda^* = \{(1, 2), (2, 1)\}$, and $\mathcal{Q}(r, \Lambda^*)$ is equal to the set of all possible $2 \times 3$ or $3 \times 2$ rectangles. It is easy to verify that in this case $|\mathcal{Q}(r, \Lambda^*)| = 2(\sqrt{n} - 1)(\sqrt{n} - 2)$. Suppose that $\Lambda = \{(1, 2)\}$. Then, $\mathcal{Q}(r, \Lambda)$ is the set of all $2 \times 3$

rectangles, and $|\mathcal{Q}(r,\Lambda)| = (\sqrt{n}-1)(\sqrt{n}-2)$. The following observations follows from the definition of $\mathcal{Q}(\cdot,\cdot)$.

LEMMA 2.8. *Let $r$ be a $d$-dimensional rectangle and for $1 \le i \le d$, let $r_i$ denote the size of $r$ along dimension $i$. Let $\Lambda \subseteq \Lambda^*$.*

$$|\mathcal{Q}(r,\Lambda)| = |\Lambda| \prod_{i=1}^{d} (\sqrt[d]{n} - r_i + 1)$$

PROOF. First, we show that for each $\lambda \in \Lambda$, $\mathcal{P}(r,\lambda)$ is still a rectangle with the length of $r_{\lambda(i)}$ along dimension $i$. Suppose that $r$ was as follows:

$$r = \{(x_1, x_2, \ldots, x_d) | \ell_i \le x_i \le \hbar_i, \quad \forall 1 \le i \le d\}$$

where $\hbar_i - \ell_i = r_i$, for all $1 \le i \le d$. From the definition of rotation, we have:

$$\mathcal{P}(r,\lambda) = \{(x_{\lambda(1)}, \ldots, x_{\lambda(d)}) | \ell_{\lambda(i)} \le x_{\lambda(i)} \le \hbar_{\lambda(i)}, \quad \forall 1 \le i \le d\}$$

Thus, we conclude that $\mathcal{P}(r,\lambda)$ is still a rectangle with the length of $r_{\lambda(i)}$ along dimension $i$. Second, we show that $|\mathcal{T}(r)| = \prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)$. Note that along each dimension $i$, $r$ has $(\sqrt[d]{n} - r_i + 1)$ different translations. Since $r$ can be translated along each dimension independently, the total number of translations should be $\prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)$. Summarizing the analysis above, we have for each $\lambda \in \Lambda$,

$$|\mathcal{T}(\mathcal{P}(r,\lambda))| = \prod_{i=1}^{d}(\sqrt[d]{n} - r_{\lambda(i)} + 1) = \prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)$$

From Definition 2.7, we have:

$$|\mathcal{Q}(r,\Lambda)| = \sum_{\lambda \in \Lambda} |\mathcal{T}(\mathcal{P}(r,\lambda))| = |\Lambda| \prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)$$

□

LEMMA 2.9. *Let $g$ be a query that is not necessarily a rectangle and for $1 \le i \le d$, let $b_i$ denote the length of the bounding box $B(g)$ along dimension $i$. Let $\Lambda \subseteq \Lambda^*$.*

$$|\mathcal{Q}(g,\Lambda)| = |\Lambda| \prod_{i=1}^{d}(\sqrt[d]{n} - b_i + 1)$$

PROOF. First, we show that for each $\lambda \in \Lambda$, the size of $B(\mathcal{P}(g,\lambda))$ should be $b_{\lambda(i)}$ along dimension $i$. For each cell $\boldsymbol{x} \in g$, let $x_i$ denote the $i$th coordinate. From the definition of bounding box, we have

$$b_i = \max_{\boldsymbol{x},\boldsymbol{y} \in g} |x_i - y_i + 1|$$

Let $\kappa_i$ be the length of $B(\mathcal{P}(g,\lambda))$ along dimension $i$. Similarly, we have $\kappa_i = \max_{\zeta,\eta \in \mathcal{P}(g,\lambda)} |\zeta_i - \eta_i + 1|$. From the definition of rotation, we have:

$$\mathcal{P}(g,\lambda) = \bigcup_{\boldsymbol{x} \in g} \{(x_{\lambda(1)}, \ldots, x_{\lambda(d)})\}$$

So we have:

$$\kappa_i = \max_{\zeta,\eta \in \mathcal{P}(g,\lambda)} |\zeta_i - \eta_i + 1| = \max_{\boldsymbol{x},\boldsymbol{y} \in g} |x_{\lambda(i)} - y_{\lambda(i)} + 1| = b_{\lambda(i)}$$

Thus from Lemma 2.8, we have for each $\lambda \in \Lambda$, $|\mathcal{T}(B(\mathcal{P}(g,\lambda)))| = \prod_{i=1}^{d}(\sqrt[d]{n} - b_i + 1)$.

Next, we observe that the number of possible translations of $g$ equals the number of possible translations of $B(g)$, so that $|\mathcal{T}(g)| = |\mathcal{T}(B(g))|$. Summarizing the analysis above, we have:

$$|\mathcal{Q}(g,\Lambda)| = \sum_{\lambda\in\Lambda} |\mathcal{T}(\mathcal{P}(g,\lambda))| = \sum_{\lambda\in\Lambda} |\mathcal{T}(B(\mathcal{P}(g,\lambda)))| = |\Lambda| \prod_{i=1}^{d}(\sqrt[d]{n} - b_i + 1)$$

□

## 3. GENERAL TECHNIQUES

Consider an arbitrary SFC $\pi$. For any two cells $\alpha, \beta \in U$ and query $q \subseteq U$, we define the function $I(q,\alpha,\beta)$ as:

$$\begin{aligned} I(q,\alpha,\beta) &= 1 \qquad \text{if } \alpha \in q \text{ and } \beta \in q \\ &= 0 \qquad \text{otherwise.} \end{aligned}$$

The SFC can be thought of as a set of directed edges that go from one cell to another, visiting each cell exactly once. Let $N(\pi)$ be the set of all such edges in SFC $\pi$, where each edge goes from a cell numbered $i$ to a cell numbered $(i+1)$, for some $0 \le i \le (n-2)$.

$$N(\pi) = \{(\pi^{-1}(i), \pi^{-1}(i+1)) | 0 \le i \le n-2\}$$

The following lemma applies to any SFC combined with any query, and gives us a powerful framework to compute the clustering number.

LEMMA 3.1. *For any query $q$ and any SFC $\pi$,*

$$c(q,\pi) = |q| - \sum_{(\alpha,\beta)\in N(\pi)} I(q,\alpha,\beta)$$

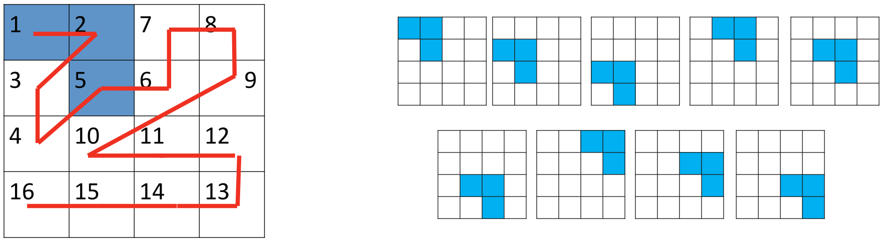
PROOF. We use proof by induction on $\sum_{(\alpha,\beta)\in N(\pi)} I(q,\alpha,\beta)$. Let $\pi(q) = \{\pi(v)|v \in q\}$. For the base case, note that the clustering number of $\pi$ for $q$ is equal to $|q|$ when no two elements in $\pi(q)$ are consecutive, since in such a case, no two elements of $q$ can belong to the same cluster. It can be seen that the cluster number will decrease by one for each pair of elements in $\pi(q)$ that are consecutive, thus forming the inductive step. □

**Example:** Consider the two dimensional $4 \times 4$ grid and SFC $\pi$ as shown in Figure 3(a). The linear order imposed by the SFC is determined by the integer assigned to each cell, shown in the upper left corner of the cell. Let $q$ be the query shown by the shaded region. Note $|q| = 3$, and $I(q,\alpha,\beta)$ is non-zero for only one pair from $N(\pi)$, which is $(\pi^{-1}(1), \pi^{-1}(2))$. Thus, we have from Lemma 3.1 that the clustering number is $c(q,\pi) = 3 - 1 = 2$, which can be verified to be correct.

For any query $q$ and a non-empty set of rotations $\Lambda$, let query set $Q = \mathcal{Q}(q,\Lambda)$. For a pair of vertices $\alpha, \beta \in U$ (perhaps non-neighboring), let $P_Q(\alpha,\beta)$ be defined as: $P_Q(\alpha,\beta) = \{r \in Q | I(r,\alpha,\beta) = 1\}$.

LEMMA 3.2.

$$c(Q,\pi) = |q| - \frac{\sum_{i=0}^{n-2} |P_Q(\pi^{-1}(i), \pi^{-1}(i+1))|}{|Q|}$$



(a) The line represents the SFC while the shaded region represents a possible query $q$.

(b) The different query regions formed by translation of $q$.

Fig. 3. An Example Set of Query Regions for an SFC

PROOF. Applying Lemma 3.1 to Definition 2.6, we have:

$$
\begin{aligned}
c(Q,\pi) &= \frac{\sum_{q\in Q} c(q,\pi)}{|Q|} \\
&= \frac{1}{|Q|}\sum_{q\in Q}\left(|q| - \sum_{(\alpha,\beta)\in N(\pi)} I(q,\alpha,\beta)\right) \\
&= |q| - \frac{1}{|Q|}\sum_{(\alpha,\beta)\in N(\pi)}\sum_{q\in Q} I(q,\alpha,\beta) \\
&= |q| - \frac{1}{|Q|}\sum_{(\alpha,\beta)\in N(\pi)} |P_Q(\alpha,\beta)| \\
&= |q| - \frac{\sum_{i=0}^{n-2} |P_Q(\pi^{-1}(i),\pi^{-1}(i+1))|}{|Q|}
\end{aligned}
$$

□

The above formula relates the clustering number $c(Q,\pi)$ to structural properties of $Q$ and $\pi$, and provides a basis for the computation of lower and upper bounds.

## 4. ANALYSIS OF A CONTINUOUS SFC FOR A RECTILINEAR QUERY

In this section, we present an exact analysis of a continuous SFC $\pi$ for any rectilinear query $g$ of a fixed size. A rectilinear query is the union of multiple disjoint $d$-dimensional rectangles. Since each cell is trivially a $d$-dimensional rectangle, and an arbitrary query can be written as the union of its constituent cells, an arbitrary query is also a rectilinear query.

From the universe $U$, we derive an undirected graph $G(U) = (U, E(U))$, whose vertex set is $U$ and where there is an edge between two vertices $v_1$ and $v_2$ in $U$ whenever $v_1$ and $v_2$ are at a Manhattan distance of $1$. For an edge $e = (v_1, v_2) \in E(U)$, we say "$e$ lies along dimension $i$" iff the coordinates of $v_1$ and $v_2$ differ along dimension $i$ (and are equal along the other dimensions). For $i = 1 \ldots d$, let $E_i(U)$ denote the subset of $E(U)$ consisting of all edges that lie along dimension $i$.

For any rectilinear query $g$, we associate a graph $G(g) = (g, E(g))$, defined as the induced subgraph of $G(U)$ with the vertex set $g$. For $i = 1 \dots d$, let $E_i(g)$ denote the set $E(g) \cap E_i(U)$, i.e. all edges in $E(g)$ that lie along dimension $i$. Note that $G(g)$ and $E_i(g)$ depend only on the query $g$, and are independent of the SFC.

### 4.1. Statement of Results

*Definition* 4.1. Given a query $g$, vector $\nu(g)$ of length $d$ is defined as: $\nu(g) = (\nu_1(g), \dots, \nu_d(g))$ where for $1 \leq i \leq d$, $\nu_i(g) = |E_i(g)|$.

The main theorem for the clustering number of a continuous SFC with respect to translations is given below.

THEOREM 4.2. **Continuous SFC, Translations Only:** *For any continuous SFC $\pi$, any query $g$ of fixed size, the average clustering number of $\pi$ for query set $\mathcal{T}(g)$ is given as:*

$$\lim_{n \to \infty} c(\mathcal{T}(g), \pi) = |g| - \mu(\pi) \cdot \nu(g)$$

*where $\cdot$ denotes the vector dot product.*

We next present the theorem when a subset of possible rotations are considered along with translations. We first introduce a new parameter for a query $g$ subject to a set of rotations $\Lambda$.

*Definition* 4.3. Given a query $g$, and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$ we define a vector $\nu(g, \Lambda)$ of length $d$ as: $\nu(g, \Lambda) = (\nu_1(g, \Lambda), \dots, \nu_d(g, \Lambda))$ where for $1 \leq i \leq d$,

$$\nu_i(g, \Lambda) = \frac{\sum_{\lambda \in \Lambda} \nu_i(\mathcal{P}(g, \lambda))}{|\Lambda|}$$

THEOREM 4.4. **Continuous SFC, Translations and Rotations:** *For any continuous SFC $\pi$, any query $g$ of a fixed size, the average clustering number of $\pi$ for query set $\mathcal{Q}(g, \Lambda)$ is given as:*

$$\lim_{n \to \infty} c(\mathcal{Q}(g, \Lambda), \pi) = |g| - \mu(\pi) \cdot \nu(g, \Lambda)$$

For example, consider an SFC $\pi_1$ shown on the left in Figure 4. Though the picture shows an $8 \times 8$ grid, the idea for an $n \times n$ grid is that the SFC goes horizontally (mostly) for the top $5n/8$ rows, and then vertically (mostly) for the bottom $3n/8$ rows. On the right are two queries $A$ and $B$, and the induced graphs $G(A)$ and $G(B)$ are shown within the queries.

By the above definitions, it can be calculated that $\mu(\pi_1) = [5/8, 3/8]$. On the right side of the figure are shown two queries $A$ and $B$. We have $\nu(A) = [2, 2]$, since $E(A)$ has $4$ edges, two of them horizontal and two vertical. From Theorem 4.2, we have the clustering number $c(\mathcal{T}(A), \pi_1) = 4 - (5/8)(2) - (3/8)(2) = 2$. Similarly, $\nu(B) = [1, 2]$. From Theorem 4.2, we have the clustering number $c(\mathcal{T}(B), \pi_1) = 4 - (5/8)(1) - (3/8)(2) = 21/8$.

### 4.2. Proofs of Theorems 4.2 and 4.4

The first part of this proof applies to a query set $Q$ constructed from a basic query $g$. It does not matter whether we construct $Q$ from translations of $g$ only, or through translations and rotations. Hence, this part will apply to proofs of both Theorems 4.2 and 4.4.

From Lemma 3.2, we have

$$c(Q, \pi) = |g| - \frac{\sum_{j=0}^{n-2} |P_Q(\pi^{-1}(j), \pi^{-1}(j+1))|}{|Q|} \tag{1}$$

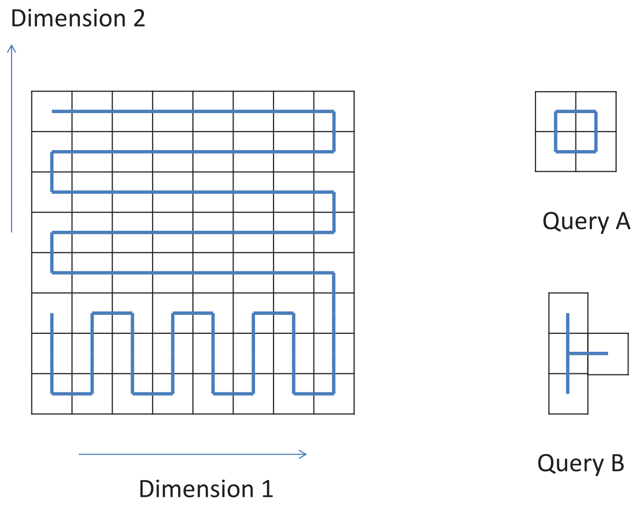


Fig. 4. On the left is an SFC $\pi_1$ and on the right are two queries $A$ and $B$.

Let $S(\cdot,\cdot)$ be defined as:

$$S(Q,\pi) = \sum_{j=0}^{n-2} |P_Q(\pi^{-1}(j), \pi^{-1}(j+1))| \tag{2}$$

Since $\pi$ is continuous, for each $j, 0 \le j \le (n-2)$, we have the pair $(\pi^{-1}(j), \pi^{-1}(j+1)) \in N^i(\pi)$ for some $i, 1 \le i \le d$. We can get the following.

$$\bigcup_{j=0}^{n-2} \{(\pi^{-1}(j), \pi^{-1}(j+1))\} = \bigcup_{i=1}^{d} N^i(\pi)$$

From the above, $S$ can be rewritten as:

$$S(Q,\pi) = \sum_{i=1}^{d} \sum_{(\alpha,\beta)\in N^i(\pi)} |P_Q(\alpha,\beta)| \tag{3}$$

We will need the following lemmas to prove Theorem 4.2. For a query set $Q$ and dimension $i, 1 \le i \le d$, let $\rho_Q^i$ be defined as:

$$\rho_Q^i = \max_{(\alpha,\beta)\in N^i(\pi)} |P_Q(\alpha,\beta)| \tag{4}$$

LEMMA 4.5. *For any $i, 1 \le i \le d$ and any query $g$,*

$$\rho^i_{\mathcal{T}(g)} \le \nu_i(g)$$

PROOF. Consider any edge $(\alpha,\beta)$ from $E_i(U)$. From the definition of $P$, we have that if query $r$ is in $P_{\mathcal{T}(g)}(\alpha,\beta)$, then $\alpha \in r$ and $\beta \in r$. Since edge $(\alpha,\beta)$ is parallel to the $i$th axis, we have the number of translations of $g$ which can include $(\alpha,\beta)$ should be no more than the number of edges in $E(g)$ which lie along dimension $i$. Thus we have:

$$|P_{\mathcal{T}(g)}(\alpha, \beta)| \leq |E_i(g)| = \nu_i(g)$$

Since the above is true for any edge $(\alpha, \beta) \in E_i(U)$, we get $\rho^i_{\mathcal{T}(g)} \leq \nu_i(g)$. □

For dimension $i, 1 \leq i \leq d$, let $N_i(\pi)$ be a subset of $N^i(\pi)$ defined as:

$$N_i(\pi) = \left\{(\alpha, \beta) \in N^i(\pi) \;\middle|\; |P_{\mathcal{T}(g)}(\alpha, \beta)| = \nu_i(g)\right\}$$

LEMMA 4.6. *For any dimension $i, 1 \leq i \leq d$, and any query $g$ of a fixed size,* $\lim_{n\to\infty} \frac{|N^i(\pi)|}{|\mathcal{T}(g)|} = \lim_{n\to\infty} \frac{|N_i(\pi)|}{|\mathcal{T}(g)|} = \mu_i(\pi)$

PROOF. Let $b_i, 1 \leq i \leq d$ be the length of $B(g)$ along dimension $i$. From Lemma 2.9, we have $\mathcal{T}(g) = \prod_{i=1}^{d}(\sqrt[d]{n} - b_i + 1)$. So from definition of $\mu_i(\pi)$, we get:

$$\lim_{n\to\infty} \frac{|N^i(\pi)|}{|\mathcal{T}(g)|} = \mu_i(\pi)$$

Now we show the second equality. Let $b^* = \max_{1\leq i\leq d} b_i$. Let $U' \subset U$ be the set of all cells $(x_1, \ldots, x_d)$ such that for each dimension $i$, $b^* - 1 \leq x_i \leq \sqrt[d]{n} - b^*$.

For any $(\alpha, \beta) \in N^i(\pi)$, if $\alpha, \beta \in U'$, then it can be seen that $|P_{\mathcal{T}(g)}(\alpha, \beta)| = |\nu_i(g)|$. The total number of pairs $(\alpha, \beta)$ such that $\alpha$ or $\beta$ lies outside of $U'$ is bounded by $n - (\sqrt[d]{n} - 2b^*)^d$. So we have:

$$|N^i(\pi)| - (n - (\sqrt[d]{n} - 2b^*)^d) \leq |N_i(\pi)| \leq |N^i(\pi)|$$

Note that

$$\lim_{n\to\infty} \frac{n - (\sqrt[d]{n} - 2b^*)^d}{\mathcal{T}(g)} = 0$$

So we have:

$$\begin{aligned}\lim_{n\to\infty} \frac{|N^i(\pi)|}{|\mathcal{T}(g)|} &= \lim_{n\to\infty} \frac{|N_i(\pi)|}{|\mathcal{T}(g)|} \\ &= \mu_i(\pi)\end{aligned}$$

□

PROOF OF THEOREM 4.2. We start from Equations 3 and 4. Setting $Q = \mathcal{T}(g)$ in Equation 3,

$$\begin{aligned}S(\mathcal{T}(g), \pi) &= \sum_{i=1}^{d} \sum_{(\alpha,\beta)\in N^i(\pi)} |P_{\mathcal{T}(g)}(\alpha, \beta)| \\ &\leq \sum_{i=1}^{d} |N^i(\pi)| \rho^i_{\mathcal{T}(g)} \quad \text{From Defn. of } \rho \\ &\leq \sum_{i=1}^{d} |N^i(\pi)| \nu_i(g) \quad \text{Using Lemma 4.5}\end{aligned}$$

Using this back in Equation 1

$$c(\mathcal{T}(g),\pi) \geq |g| - \frac{\sum_{i=1}^{d} |N^i(\pi)|\nu_i(g)}{|\mathcal{T}(g)|} \tag{5}$$

Taking limits on the right side and applying Lemma 4.6:

$$\begin{aligned}\lim_{n\to\infty} c(\mathcal{T}(g),\pi) &\geq |g| - \lim_{n\to\infty}\sum_{i=1}^{d}\frac{|N^i(\pi)|}{|\mathcal{T}(g)|}\nu_i(g)\\ &= |g| - \sum_{i=1}^{d}\mu_i(\pi)\nu_i(g)\end{aligned}$$

We now consider the upper bound on $c(\mathcal{T}(g),\pi)$. The starting point for this is Equations 3 and 4. Using Equation 3

$$S(\mathcal{T}(g),\pi) \geq \sum_{i=1}^{d}|N_i(\pi)|\nu_i(g)$$

Proceeding as above,

$$c(\mathcal{T}(g),\pi) \leq |g| - \frac{\sum_{i=1}^{d} |N_i(\pi)|\nu_i(g)}{|\mathcal{T}(g)|} \tag{6}$$

Taking limits on both sides and applying Lemma 4.6

$$\lim_{n\to\infty} c(\mathcal{T}(g),\pi) \leq |g| - \sum_{i=1}^{d}\mu_i(\pi)\nu_i(g)$$

This upper bound on the clustering number, when combined with the lower bound, completes the proof. □

PROOF OF THEOREM 4.4.

$$P_Q(\pi^{-1}(j),\pi^{-1}(j+1)) = \bigcup_{\lambda\in\Lambda} P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))$$

Since the different $P_Q(\lambda)$s are disjoint for different $\lambda$

$$\begin{aligned}S(Q,\pi) &= \sum_{j=0}^{n-2}|P_Q(\pi^{-1}(j),\pi^{-1}(j+1))|\\ &= \sum_{j=0}^{n-2}\sum_{\lambda\in\Lambda}|P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))|\\ &= \sum_{\lambda\in\Lambda}\sum_{j=0}^{n-2}|P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))|\end{aligned}$$

From Equation 1, we have:

$$
\begin{aligned}
c(\mathcal{Q}(g,\Lambda),\pi) &= |g| - \frac{S(\mathcal{Q}(g,\Lambda),\pi)}{|\mathcal{Q}(g,\Lambda)|} \\
&= |g| - \frac{1}{|\Lambda||Q(\lambda)|}\sum_{\lambda\in\Lambda}\sum_{j=0}^{n-2}|P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))| \\
&= \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}\left[|g| - \frac{\sum_{j=0}^{n-2}|P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))|}{|Q(\lambda)|}\right]
\end{aligned}
$$

Applying Theorem 4.2:

$$\lim_{n\to\infty} c(\mathcal{Q}(g,\Lambda),\pi) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}(|g| - \mu(\pi)\cdot\nu(\mathcal{P}(g,\lambda)))$$

After simplification, we have:

$$\lim_{n\to\infty} c(\mathcal{Q}(g,\Lambda),\pi) = |g| - \mu(\pi)\cdot\nu(g,\Lambda)$$

□

### 4.3. All Possible Rotations

When $\Lambda = \Lambda^*$, the set of all possible rotations, we get a much simpler form for the clustering number of a continuous SFC, as follows. For any query $q \subset U$ such that $q$ does not contain a cell adjacent to the boundary of $U$ (i.e. $q$ does not have any cell with a coordinate equal to $0$ or $\sqrt[d]{n}-1$) the surface area of $q$ is defined to be the number of cells $\beta \in U$ such that $\beta \notin q$, and $\beta$ is at a Manhattan distance of $1$ from some cell $\alpha$ in $q$. For a query $q$ that has at least one cell on the boundary of $U$, we add for each such cell, the number of its coordinates that are equal to $0$, or $\sqrt[d]{n}-1$.

LEMMA 4.7. *The surface area of $g$ is $S_g = 2d|g| - 2|E(g)|$.*

PROOF. For $\alpha \in g$, let $\omega(\alpha)$ denote the degree of $\alpha$ in $G(g)$. The contribution of $\alpha$ to the surface area is $2d - \omega(\alpha)$. Thus, the total surface area of $g$ is: $\sum_{\alpha\in g}[2d-\omega(\alpha)] = 2d|g| - \sum_{\alpha\in g}\omega(\alpha)$. The lemma follows by noting that for a graph the sum of degrees is twice the number of edges. □

THEOREM 4.8. *For any continuous SFC $\pi$, and any query $g$ of a fixed size,*

$$\lim_{n\to\infty} c(\mathcal{Q}(g,\Lambda^*),\pi) = \frac{S_g}{2d}$$

PROOF. From Theorem 4.4, we have

$$\lim_{n\to\infty} c(\mathcal{Q}(g,\Lambda^*),\pi) = |g| - \mu(\pi)\cdot\nu(g,\Lambda^*)$$

Note that for $1 \le i \le d$,

$$\nu_i(g,\Lambda^*) = \frac{\sum_{\lambda\in\Lambda^*}|E_i(\mathcal{P}(g,\lambda))|}{|\Lambda^*|}$$

Let $e_i = \sum_{\lambda\in\Lambda^*}|E_i(\mathcal{P}(g,\lambda))|$. When all the $d!$ possible rotations are considered, by symmetry we have $e_1 = e_2 = \ldots = e_d$. Further, $\sum_{i=1}^{d} e_i = |E(g)|d!$. Thus, we have $e_i = \frac{|E(g)|d!}{d}$ for each $i, 1 \le i \le d$. $\nu_i(g,\Lambda^*) = \frac{|E(g)|}{d}$

$$\begin{aligned} A &= |g| - \mu(\pi)\left[\frac{|E(g)|}{d}, \frac{|E(g)|}{d}, \ldots, \frac{|E(g)|}{d}\right] \\ &= |g| - \frac{|E(g)|}{d} \text{ since } \textstyle\sum_{i=1}^{d} \mu_i(\pi) = 1 \\ &= \frac{S_g}{2d} \text{ using Lemma 4.7} \end{aligned}$$

□

Since the Hilbert curve is a continuous curve, the result of Moon *et al.* [Moon et al. 2001] follows from the above theorem.

### 4.4. Symmetric SFCs

We say that a continuous SFC $\pi$ is *symmetric* if it has (nearly) the same number of edges along each dimension $i$. More precisely, we need that for each $i = 1 \ldots d$, $\mu_i(\pi)$ exists and is equal to $\frac{1}{d}$.

COROLLARY 4.9. *For any symmetric SFC $\pi$, for any query $g$ of a fixed size, for any non-empty set of rotations $\Lambda \subseteq \Lambda^*$*

$$\lim_{n\to\infty} c(\mathcal{Q}(g, \Lambda), \pi) = \frac{S_g}{2d}$$

PROOF. The proof follows from Theorem 4.4, and then using a similar technique used in the proof of Theorem 4.8. The difference being that in Theorem 4.8 the vector $\nu(g, \Lambda)$ had all elements equal, while in this case the vector $\mu(\pi)$ has all elements equal. □

It is known that the $d$-dimensional Hilbert curve $\mathcal{H}_d$ is symmetric (see [Moon et al. 2001], Section 3). From the above corollary, it follows that Hilbert curve yields the same performance for a query irrespective of the set of rotations considered.

## 5. RECTANGULAR QUERIES: LOWER BOUND FOR ANY SFC

In this section, we present a lower bound on the clustering number of any SFC, for rectangular queries. Further, we show that *for a query set formed by translation and/or rotations of a rectangular query, there exists a continuous SFC that is optimal.*

### 5.1. Statement of Results

Consider the query set $\mathcal{Q}(r, \Lambda)$, where $r$ is a rectangular query, and $\Lambda \subseteq \Lambda^*$ is a non-empty set of rotations. Let $\nu^{max} = \nu^{max}(r, \Lambda) = \max_{1\le i\le d} \nu_i(r, \Lambda)$. The main results in this section are stated in Theorems 5.1 and 5.2.

THEOREM 5.1. *Given a rectangular query $r$ of a fixed size and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$, for any SFC $\pi$ (not necessarily continuous), if $\lim_{n\to\infty} c(\mathcal{Q}(r, \Lambda), \pi)$ exists, then*

$$\lim_{n\to\infty} c(\mathcal{Q}(r, \Lambda), \pi) \ge |r| - \nu^{\max}$$

THEOREM 5.2. *For a rectangular query $r$ of a fixed size and $\Lambda \subseteq \Lambda^*$, there exists a continuous SFC $\pi$ whose clustering number is optimal, i.e.:*

$$\lim_{n\to\infty} c(\mathcal{Q}(r, \Lambda), \pi) = |r| - \nu^{\max}$$

We also have the following surprising fact, that when the set of all rotations are considered for a rectangular query, every continuous SFC $\pi$ is optimal.

COROLLARY 5.3. *For a rectangular query $r$ of a fixed size, if all possible rotations are considered, then* any *continuous SFC $\pi$ is optimal.*

### 5.2. Proofs of Theorems 5.1 and 5.2

To prove Theorem 5.1, we need the following lemma. Let $r$ be a rectangular query of a fixed size and $\Lambda \subseteq \Lambda^*$ be a non-empty set of rotations. Set $Q = \mathcal{Q}(r, \Lambda)$.

LEMMA 5.4. *For any SFC $\pi$ and any pair $(\alpha, \beta) \in N(\pi)$, we have:*

$$|P_Q(\alpha, \beta)| \le |\Lambda|\nu^{max}$$

PROOF. Let $\gamma = \{\alpha, \beta\}$ and $B(\gamma)$ be the bounding box of $\gamma$. For each $q \in P_Q(\alpha, \beta)$, we have $B(\gamma) \subseteq q$ since $q$ is a rectangle.

Note that $B(\gamma)$ is also a rectangle. Thus we can always secure a neighboring cell of $\alpha$, say $\alpha'$, such that $\alpha' \in B(\gamma)$ and the Manhattan distance between $\alpha$ and $\alpha'$ is 1.

Let $\gamma' = \{\alpha, \alpha'\}$. Since $\gamma' \subseteq q$, we have $q \in P_Q(\gamma')$. Thus we have that $P_Q(\gamma) \subseteq P_Q(\gamma')$.

$$|P_Q(\gamma)| \le |P_Q(\gamma')| \tag{7}$$

For $\lambda \in \Lambda$, let $Q(\lambda) = \mathcal{T}(\mathcal{P}(r, \lambda))$. Note that:

$$|P_Q(\gamma')| = \sum_{\lambda \in \Lambda} |P_{Q(\lambda)}(\gamma')| \tag{8}$$

Assume $\gamma'$ is parallel to the $i$th axis. For any $\lambda \in \Lambda$, we have the following:

$$|P_{Q(\lambda)}(\gamma')| \le \nu_i(\mathcal{P}(r, \lambda)) \tag{9}$$

The above can be proved using an argument identical to the one used in Lemma 4.5. In Lemma 4.5, this was used to bound the size of $P_Q(\alpha'', \beta'')$ where $\alpha''$ and $\beta''$ are neighbors in a continuous SFC, but this exact argument can be used here too since $\alpha$ and $\alpha'$ are at a Manhattan distance of 1.

Combining Equations 7, 8, and 9,

$$|P_Q(\gamma)| \le \sum_{\lambda \in \Lambda} \nu_i(\mathcal{P}(r, \lambda)) = |\Lambda|\nu_i(r, \Lambda) \le |\Lambda|\nu^{max}$$

□

PROOF OF THEOREM 5.1. From Lemma 3.2, we have:

$$c(Q, \pi) = |r| - \frac{1}{|Q|} \sum_{(\alpha,\beta) \in N(\pi)} |P_Q(\alpha, \beta)|$$

Let $r_i, 1 \le i \le d$ denote the length of $r$ along dimension $i$. Applying Lemma 2.8, we have:

$$c(\mathcal{Q}(r, \Lambda), \pi) = |r| - \frac{1}{|\Lambda| \prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)} \sum_{(\alpha,\beta) \in N(\pi)} |P_Q(\alpha, \beta)|$$

Applying Lemma 5.4:

$$\begin{aligned} c(\mathcal{Q}(r, \Lambda), \pi) &\ge |r| - \frac{1}{|\Lambda| \prod_{i=1}^{d}(\sqrt[d]{n} - r_i + 1)}(n-1)|\Lambda|\nu^{max} \\ &= |r| - \nu^{max} - o(1) \end{aligned}$$

In the above, we use $o(1)$ to denote a function of $n$ that approaches $0$ as $n \to \infty$. The proof depends on the fact $\lim_{n\to\infty} \frac{n-1}{\prod_{i=1}^{d}(\sqrt[d]{n}-r_i+1)} = 1$ which is true since $r_i$ and $d$ are constants independent of $n$. □

Proof of Theorem 5.2. We present an algorithm which can construct a continuous SFC whose performance meets the lower bound from Theorem 5.1. Let $j = \text{argmax}_{1\le i\le d}\, \nu_i(r,\lambda)$, so that $\nu_j = \nu^{max}$. Partition all cells in $U$ into $(\sqrt[d]{n})^{d-1}$ disjoint groups:

$$L_{x'_1,\cdots,x'_{j-1},x'_{j+1},\cdots,x'_d} = \{(x'_1,\cdots,x'_{j-1},x_j,x'_{j+1},\cdots,x'_d) | 0 \le x_j \le \sqrt[d]{n}-1\}$$

where $(x'_1,\cdots,x'_{j-1},x'_{j+1},\cdots,x'_d)$ are parameters and each of them ranges from $0$ to $\sqrt[d]{n}-1$. We observe that the cells in each group form a $1$-dimensional line aligning with the $j$th dimension. The idea for constructing an SFC with the optimal clustering number for the set of queries $\mathcal{Q}(r,\Lambda)$ is to simply make sure that all points in the set $L_{x'_1,\cdots,x'_{j-1},x'_{j+1},\cdots,x'_d}$ are ordered consecutively by the SFC; the ordering across sets does not matter.

The following SFC $S_j$ satisfies the above criteria.

$$S^j((x_1,\ldots,x_d)) = \sum_{i=1}^{j-1} x_i(\sqrt[d]{n})^i + x_j + \sum_{i=j+1}^{d} x_i(\sqrt[d]{n})^{i-1}$$

We can check that:

$$\mu_j(S^j) = 1, \quad \mu_i(S^j) = 0, \forall i \ne j$$

From Theorem 4.4, we get that for any rectilinear query $r$ and rotation set $\Lambda$,

$$\lim_{n\to\infty} c(\mathcal{Q}(r,\Lambda), S^j) = |r| - \nu^{max}$$

So from Theorem 5.1, we conclude that for any rectangle query $r$ and any nonempty set of rotations $\Lambda \subseteq \Lambda^*$, $S^j$ is optimal among all SFCs. □

The following algorithm summarizes the method for deriving an optimal SFC given a rectangular query and a subset of rotations.

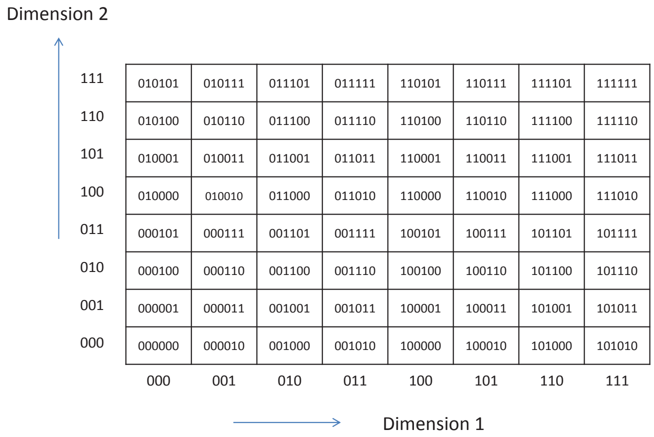
```
**Algorithm 1:** Optimal SFC for a Rectangular Query
**Input**: Rectangular query $r$, subset of rotations $\Lambda \subseteq \Lambda^*$.
**Output**: SFC $\pi$ with optimal clustering number for set $\mathcal{Q}(r,\Lambda)$.
Let $j = \text{argmax}_{1\le i\le d}\, \nu_i(r,\Lambda)$
For cell $(x_1,x_2,\ldots,x_d) \in U$, $\pi(x) = \sum_{i=1}^{j-1} x_i(\sqrt[d]{n})^i + x_j + \sum_{i=j+1}^{d} x_i(\sqrt[d]{n})^{i-1}$
```

Proof of Corollary 5.3. Consider a continuous SFC $\pi$. Using Theorem 4.8,

$$c(\mathcal{Q}(r,\Lambda^*),\pi) = \frac{S_r}{2d}$$

where $S_r$ denotes the surface area of $r$. If $\Lambda = \Lambda^*$, then for $i = 1\ldots d$, $\nu^{max} = \nu_i(r,\Lambda^*)$, and thus $\nu^{max} = \frac{|E(r)|}{d}$.

The lower bound from Theorem 5.1 is $|r| - \nu^{\max} = |r| - \frac{|E(r)|}{d}$. Proceeding similarly to the proof of Theorem 4.8, we get the above expression to be $\frac{S_r}{2d}$. Thus, the performance of $\pi$ meets the lower bound, showing that it is optimal. □



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 111 | 010101 | 010111 | 011101 | 011111 | 110101 | 110111 | 111101 | 111111 |
| 110 | 010100 | 010110 | 011100 | 011110 | 110100 | 110110 | 111100 | 111110 |
| 101 | 010001 | 010011 | 011001 | 011011 | 110001 | 110011 | 111001 | 111011 |
| 100 | 010000 | 010010 | 011000 | 011010 | 110000 | 110010 | 111000 | 111010 |
| 011 | 000101 | 000111 | 001101 | 001111 | 100101 | 100111 | 101101 | 101111 |
| 010 | 000100 | 000110 | 001100 | 001110 | 100100 | 100110 | 101100 | 101110 |
| 001 | 000001 | 000011 | 001001 | 001011 | 100001 | 100011 | 101001 | 101011 |
| 000 | 000000 | 000010 | 001000 | 001010 | 100000 | 100010 | 101000 | 101010 |
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

Fig. 5. The assignment of keys to cells in a two dimensional $Z$ curve on an $8 \times 8$ grid.

## 6. PERFORMANCE OF $Z$ CURVE

In this section, we consider the performance of $Z$ curve (see Figure 1(c)), which is possibly the most popular non-continuous SFC, and compare it to the lower bound derived for continuous SFCs. To simplify our analysis, we assume the query $q$ is a $d$-dimensional cube with size $m \times \cdots \times m$ and the query set is $\mathcal{T}(q)$, i.e. all possible translation of $q$ in $U$. The $Z$ curve on $d$ dimensions is usually defined for a universe whose side length is a power of two. Let $\sqrt[d]{n} = 2^k$. A cell $x$ in the universe $U$ can be represented by coordinates $(x_1, x_2, \ldots, x_d)$, where for $1 \leq i \leq d$, $0 \leq x_i < 2^k$; thus $x_i$ can be represented using a binary string of length $k$. For $j = 1 \ldots d$, let $x_i^j$ denote the $j$-th most significant bit in $x_i$.

We recall the definition of a $d$ dimensional $Z$ curve ([Orenstein and Merrett 1984; Morton 1966]). The $Z$ curve is defined by assigning to each cell $x \in U$, a "key" $Z(x)$, which is an integer that denotes the position of a cell in the space filling curve order. *$Z(x)$ is equal to the binary number represented by the string* $x_1^1, x_2^1, \cdots, x_d^1, x_1^2, x_2^2, \cdots, x_d^2, \cdots, x_1^k, x_2^k, \cdots, x_d^k$. In other words, the coordinates in different dimensions are interleaved together to form the key of the cell. For example, if $d = 3, k = 3$, then $Z(101, 010, 011) = 100011101$. Figure 5 illustrates how a two dimensional $Z$ curve assigns keys to each cell on an $8 \times 8$ grid. Note that different $Z$ curves are possible by taking the dimensions in a different order during interleaving, but these are all equivalent to the above definition, at least for the clustering number that we consider.

THEOREM 6.1. *Let $Z$ be the $d$-dimensional Z-curve and $q$ the cube of size $m^d$. Then:*

$$\lim_{n\to\infty} c(\mathcal{T}(q), Z) = \left(2 + o\left(\frac{1}{2^d}\right)\right) m^{d-1} + o\left(m^{d-1}\right)$$

From Theorem 5.1, the lower bound on $\lim_{n\to\infty} c(\mathcal{T}(q), Z)$ is $m^{d-1}$. Thus we conclude that the performance of the $Z$ curve is a factor of two away from the lower bound.

Our strategy is to use Lemma 3.2. We first partition $N(Z)$ into $kd$ disjoint parts. Then we calculate $|P_{\mathcal{T}(q)}(\alpha, \beta)|$ when $\alpha, \beta$ falls into each specific part and sum them up together. Recall $N(Z) = \{(Z^{-1}(j), Z^{-1}(j+1)) | 0 \leq j \leq n-2\}$. We divide the $N(Z)$ into

$kd$ groups $G_h, 0 \le h \le kd-1$ as follows. Let $G_h = \{(Z^{-1}(j), Z^{-1}(j+1))\}$ where $j$ is equal to a binary number which has the form of $(*, *, \cdots, 0, 1, 1, h \text{ times}, 1)$. That is to say $j$ can be denoted by a binary string such that the $h$ least significant bits of the string are 1 while the $h+1$ least significant of the string is 0. We can see that $G_h, 0 \le h \le kd-1$ is a disjoint partition of $N(Z)$. For each $0 \le h \le kd-1$, let $|G_h|$ be the number of elements in $G_h$.

LEMMA 6.2.

$$|G_h| = 2^{kd-h-1} = \frac{n}{2^{h+1}}$$

PROOF. Note that there are $(kd-h-1)$ free bits in the binary string of $j$, resulting in $2^{(kd-h-1)}$ possible different values that $j$ can take in $G_h$. □

Assume $h$ can be uniquely represented as $h = k_1 d + k_2$ where $0 \le k_1 \le k-1, 0 \le k_2 \le d-1, k_1 \in \mathbb{Z}, k_2 \in \mathbb{Z}$. Let $(\alpha, \beta) \in G_h$, where $0 \le h \le kd-1$.

The following lemma shows that the difference in coordinates of $\alpha$ and $\beta$ along each dimension is determined by $k_1$ and $k_2$ exclusively. Let $\alpha = (\alpha_1, \cdots, \alpha_d), \beta = (\beta_1, \cdots, \beta_d)$.

LEMMA 6.3.

$$\alpha_i - \beta_i = \begin{cases} 2^{k_1+1} - 1 & \textit{if } d - k_2 + 1 \le i \le d \\ -1 & \textit{if } i = d - k_2 \\ 2^{k_1} - 1 & \textit{if } 1 \le i \le d - k_2 - 1 \end{cases}$$

PROOF. According to the definition of $G_h$, we get to know $(\alpha, \beta) = (Z^{-1}(j), Z^{-1}(j+1))$ for some $i$ where the binary expressions of $j$ and $j+1$ have a special structure as follows:

$$\begin{aligned} j &= (\underbrace{*, \cdots, *}_{(k-k_1-1)d}, \underbrace{\overbrace{*, \cdots, *}^{d-k_2-1}, 0, \overbrace{1, \cdots, 1}^{k_2}}_{d}, \underbrace{1, \cdots, 1}_{k_1 d}) \\ j+1 &= (\underbrace{*, \cdots, *}_{(k-k_1-1)d}, \underbrace{\overbrace{*, \cdots, *}^{d-k_2-1}, 1, \overbrace{0, \cdots, 0}^{k_2}}_{d}, \underbrace{0, \cdots, 0}_{k_1 d}) \end{aligned}$$

For the $d$-dimensional $Z$ curve, we get for $d - k_2 + 1 \le i \le d$,

$$\begin{aligned} \alpha_i &= (\overbrace{*, \cdots, *}^{k-k_1-1}, \overbrace{1, \cdots, 1}^{k_1+1}) \\ \beta_i &= (\overbrace{*, \cdots, *}^{k-k_1-1}, \overbrace{0, \cdots, 0}^{k_1+1}) \end{aligned}$$

Similarly, for the case $i = d - k_2$ and $1 \le i \le d - k_2 - 1$, $\alpha_i$ and $\beta_i$ can be expressed respective as follows:

$$\left( \begin{aligned} & i = d - k_2 \\ & \alpha_i = (\overbrace{*, \cdots, *}^{k-k_1-1}, 0, \overbrace{1, \cdots, 1}^{k_1}) \\ & \beta_i = (\overbrace{*, \cdots, *}^{k-k_1-1}, 1, \overbrace{0, \cdots, 0}^{k_1}) \end{aligned} \right), \quad \left( \begin{aligned} & 1 \le i \le d - k_2 - 1 \\ & \alpha_i = (\overbrace{*, \cdots, *}^{k-k_1}, \overbrace{1, \cdots, 1}^{k_1}) \\ & \beta_i = (\overbrace{*, \cdots, *}^{k-k_1}, \overbrace{0, \cdots, 0}^{k_1}) \end{aligned} \right).$$

In each case, we can verify the difference between $\alpha_i$ and $\beta_i$ just as indicated as Lemma 6.3. □

For each pair $(\alpha, \beta) \in G_h$, the following lemma gives a formula for computing $|P_{\mathcal{T}(q)}(\alpha, \beta)|$. Let $B(\alpha, \beta)$ be the bounding box of the query $\{\alpha, \beta\}$. We can verify that the length of $B(\alpha, \beta)$ along the $i$th dimension should be $b_i = |\alpha_i - \beta_i| + 1$ for each $1 \le i \le d$. Let $b^* = \max_{1 \le i \le d} b_i$. Let $U' \subset U$ be the set of all cells $(x_1, \cdots, x_d)$ such that for each dimension $i$, $m - 1 \le x_i \le \sqrt[d]{n} - m$.

LEMMA 6.4. *Suppose* $\{\alpha, \beta\} \subseteq U'$ *and* $b^* \le m$. *Then:*

$$|P_{\mathcal{T}(q)}(\alpha, \beta)| = \prod_{i=1}^{d} (m - |\alpha_i - \beta_i|)$$

PROOF. We first show that for any $g \in \mathcal{T}(q)$,

$$(\alpha, \beta) \subseteq g \Longleftrightarrow B(\alpha, \beta) \subseteq g.$$

Suppose $(\alpha, \beta) \subseteq g$. Since $B(\alpha, \beta)$ is the smallest rectangle including both of $\alpha$ and $\beta$ while $g$ is a cube, we have $B(\alpha, \beta) \subseteq g$. The arrow from right to left is obvious since $\{\alpha, \beta\} \subseteq B(\alpha, \beta)$. Thus we conclude that $|P_{\mathcal{T}(q)}(\alpha, \beta)|$ should be equal to the number of queries in $\mathcal{T}(q)$ that includes $B(\alpha, \beta)$.

Note that $\{\alpha, \beta\} \subseteq U'$, which implies $B(\alpha, \beta)$ lies in a relative central part of the universe. Thus we can view the problem of how many different translations of $q$ can contain $B(\alpha, \beta)$ as the problem of how many different positions we can put $B(\alpha, \beta)$ into $q$. According to Lemma 2.8, we reach our conclusion. □

Let $K = \lfloor \log_2 m \rfloor$. Also assume for $0 \le h \le kd-1$, for any pair $(\alpha, \beta) \in G_h$, $\{\alpha, \beta\} \subseteq U'$. Note that even though the assumption $\{\alpha, \beta\} \subseteq U'$ is not true in all cases, it will not affect the result in Theorem 6.1, as we explain further.

LEMMA 6.5.

$$\sum_{(\alpha,\beta)\in G_h} |P_Q(\alpha, \beta)| = \begin{cases} \frac{n}{2^{(k_1 d + k_2 + 1)}} (m-1)(m - 2^{k_1+1} + 1)^{k_2} (m - 2^{k_1} + 1)^{d-1-k_2} & \textit{if } h \le Kd \\ 0 & \textit{if } h > Kd \end{cases}$$

PROOF. We can see that when $h > Kd$, the largest size of the bounding box of any pair $(\alpha, \beta) \in G_h$ will exceed $m$, thus none of queries could include $(\alpha, \beta)$. Suppose $h \le Kd$. From Lemma 6.2, we get there are totally $\frac{n}{2^{h+1}}$ pairs in $G_h$ while from Lemmas 6.3 and 6.4, we have that for each pair $(\alpha, \beta) \in G_h$, there are $(m-1)(m - 2^{k_1+1} + 1)^{k_2}(m - 2^{k_1} + 1)^{d-1-k_2}$ different queries that include $(\alpha, \beta)$. □

PROOF. We now prove Theorem 6.1. Let $Q = \mathcal{T}(q)$. From Lemma 3.2, we have:

$$c(Q, Z) = |q| - \frac{1}{|Q|} \sum_{(\alpha,\beta)\in N(Z)} |P_Q(\alpha, \beta)| \tag{10}$$

Let $S(Q, Z) = \sum_{(\alpha,\beta)\in N(Z)} |P_Q(\alpha, \beta)|$. According to the definition of $G_h, 0 \le h \le kd-1$, we can rewrite $S(Q, Z)$ as follows:

$$S(Q, Z) = \sum_{h=0}^{kd-1} \sum_{(\alpha,\beta)\in G_h} |P_Q(\alpha, \beta)| \tag{11}$$

First we justify our assumption that for each $(\alpha, \beta) \in N(Z)$, $(\alpha, \beta) \subseteq U'$ will make no difference in the result of Theorem 6.1. Let $\widehat{N}(Z) \subseteq N(Z)$ such that for each element $(\alpha, \beta) \in \widehat{N}(Z)$, we have $\alpha \in U - U'$ or $\beta \in U - U'$. We can show that $|\widehat{N}(Z)| \le 2|U - U'| = O(n^{1-\frac{1}{d}})$. According to Lemma 5.4, we have $|P_Q(\alpha, \beta)| \le (m-1)m^{d-1}$ for any

$(\alpha, \beta) \in N(Z)$. Let $\Delta S(Q, Z)$ be the absolute value of difference brought to $S(Q, Z)$ after we assume all cells fall in $U'$. Therefore we have that:

$$\Delta S(Q, Z) \leq 2(m-1)m^{d-1}O(n^{1-\frac{1}{d}}) = o(n)$$

Note that $|Q| = (\sqrt[d]{n} - m + 1)^d = \Theta(n)$. Thus we can conclude that our assumption will make no difference to $c(Q, Z)$ in 10 when $n$ approaches to infinity.

Let $h = k_1 d + k_2$ where $0 \leq k_1 \leq k-1, 0 \leq k_2 \leq d-1$. Combining Lemmas 6.2 and 6.5, we get:

$$\begin{aligned} S(Q, Z) &= \sum_{h=0}^{Kd} \sum_{(\alpha,\beta) \in G_h} |P_Q(\alpha, \beta)| && (12) \\ &= \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{d-1} \frac{n}{2^{k_1 d + k_2 + 1}} (m-1)(m - 2^{k_1+1} + 1)^{k_2} (m - 2^{k_1} + 1)^{d-1-k_2} && (13) \\ &+ \frac{n}{2^{Kd+1}} (m-1)(m - 2^K + 1)^{d-1} && (14) \end{aligned}$$

We are interested in computing $\lim_{n\to\infty} \frac{S(Q,Z)}{|Q|}$, and examine each term in the above expression.

We first consider the term in (14). Recall that $K = \lfloor \log_2 m \rfloor$, so that $1 \leq \frac{m}{2^K} \leq 2$. When $m$ goes to infinity, this term is bounded by $\frac{nm^d}{2 \times 2^{Kd}}$, which is further bounded by $n2^{d-1}$. Finally, when divided by $|Q|$ (which is nearly $n$), the contribution of the above term is $O(2^{d-1})$, which is certainly $o(m^{d-1})$.

From (13), we find that the coefficient of $m^d$ in $\lim_{n\to\infty} \frac{S(Q,Z)}{|Q|}$ is:

$$\sum_{k_1=0}^{K-1} \sum_{k_2=0}^{d-1} \frac{1}{2^{k_1 d + k_2 + 1}} = \sum_{k_1=0}^{K-1} \frac{1}{2^{k_1 d}} \sum_{k_2=0}^{d-1} \frac{1}{2^{k_2+1}} = (1 + 2^{-d} + \cdots + 2^{-d(K-1)})(1 - 2^{-d}) = (1 - 2^{-dK})$$

The coefficient of $m^{d-1}$ in $\lim_{n\to\infty} \frac{S(Q,Z)}{|Q|}$ is:

$$(-1) \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{d-1} \frac{1}{2^{k_1 d + k_2 + 1}} \left(1 + (2^{k_1+1} - 1)k_2 + (2^{k_1} - 1)(d - 1 - k_2)\right) = (-1)(2 + o(2^{-d}))$$

Therefore we get:

$$\lim_{n\to\infty} \frac{S(Q, Z)}{|Q|} = m^d - (2 + o(2^{-d}))m^{d-1} + o(m^{d-1}). \qquad (15)$$

Substituting Equation 15 into 10 we reach Theorem 6.1. □

## 7. EXTENSIONS

In this section, we present extensions and generalizations to our analysis of clustering properties of SFCs. In Section 7.1 we present an analysis of "near-continuous" SFCs, which are a broader class than continuous SFCs as defined here. In Section 7.2, we consider a natural class of queries called "connected queries", and present some observations on the clustering numbers for such queries. In Section 7.3 we show that our results on rectangular queries do not change if we consider simple union of shapes produced by rotations versus a multiset union, as defined in Definition 2.7.

### 7.1. Near-continuous SFCs

We consider a class of SFCs that we call near-continuous SFCs, which are a more general class of SFCs than the class of continuous SFCs, defined in Section 2.2. A near-continuous SFC is required to move move from one cell in the multidimensional space to another cell which can be "diagonally" connected. More precisely, for two cells $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$, let $\Delta(x, y) = \max_{1 \leq i \leq d} |x_i - y_i|$.

*Definition* 7.1. An SFC $\pi$ is said to be a near-continuous SFC if it has the property that for every $0 \leq i \leq n-2$, $\Delta(\pi^{-1}(i), \pi^{-1}(i+1)$ is 1.

Our definition of a near-continuous SFC corresponds to the definition of SFCs that are called "continuous" in the work of [Bugnion et al. 1997], though they only consider curves in two dimensions. We first observe that each continuous SFC is also a near-continuous SFC, but there are SFCs which have been invented that are near-continuous, but not continuous; for instance, the SFC due to Asano *et al.* [Asano et al. 1997].

Let $\pi$ be a near-continuous SFC. Consider the set of vectors $V' = \{0, -1, +1\}^d - (0, 0, \ldots, 0)$. Clearly, $V'$ has $3^d - 1$ elements, and represents all possible "directions" along which an edge of $N(\pi)$ could lie. The elements of $V'$ are paired into a set $V$ of $(3^d - 1)/2$ vectors as follows: for every pair of vectors $v_1, v_2 \in V'$ such that $v_1 + v_2 = 0$, only one of $v_1$ or $v_2$ is arbitrarily chosen into $V$.

Let $\bar{d} = (3^d - 1)/2$. We arbitrarily number the vectors in $V$ from 1 till $\bar{d}$. For edge $(\alpha, \beta) \in N(\pi)$, we have $\alpha - \beta \in V'$, and the edge lies along a unique vector of $V$. For $i = 1 \ldots \bar{d}$, let $N^i(\pi)$ denote the set of edges of $\pi$ that lie along the $i$th vector of $V$. We extend the definition of vector $\mu$ from continuous SFCs to near-continuous SFCs as follows.

*Definition* 7.2. For a near-continuous SFC $\pi$, $\mu(\pi)$ is a vector of length $\bar{d}$. $\mu(\pi) = (\mu_1(\pi), \mu_2(\pi), \ldots, \mu_{\bar{d}}(\pi))$, where for $i = 1 \ldots \bar{d}$

$$\mu_i(\pi) = \lim_{n \to \infty} \frac{|N^i(\pi)|}{n-1}$$

We assume the vector $\mu(\pi)$ exists for all near-continuous SFCs that we consider. For query $g$, we also define the vector $\nu^{'}(g)$ for a near-continuous SFC as follows. For query $g$, $\nu^{'}(g)$ is a vector with $\bar{d}$ elements $(\nu^{'}_1(g), \nu^{'}_2(g), \ldots, \nu^{'}_{\bar{d}}(g))$, where for $i = 1 \ldots \bar{d}$, $\nu^{'}_i(g)$ is the number of positions within $g$ where the $i$th vector in $V$ can be placed, so that both endpoints of the vector lie within $g$. In other words, $\nu^{'}_i(g)$ is the number of cells $\alpha \in g$ such that $\alpha + V(i) \in g$, where $V(i)$ is the $i$th vector of $V$.

*Definition* 7.3. Given a query $g$, and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$ we define a vector $\nu^{'}(g, \Lambda)$ of length $\bar{d}$ as: $\nu^{'}(g, \Lambda) = (\nu^{'}_1(g, \Lambda), \ldots, \nu^{'}_{\bar{d}}(g, \Lambda))$ where for $1 \leq i \leq \bar{d}$,

$$\nu^{'}_i(g, \Lambda) = \frac{\sum_{\lambda \in \Lambda} \nu^{'}_i(\mathcal{P}(g, \lambda))}{|\Lambda|}$$

Similar to Theorem 4.4, we have:

THEOREM 7.4. **Near-Continuous SFC, Translations and Rotations:** *For any near-continuous SFC $\pi$, any query $g$ of a fixed size, the average clustering number of $\pi$ for query set $\mathcal{Q}(g, \Lambda)$ is given as:*

$$\lim_{n \to \infty} c(\mathcal{Q}(g, \Lambda), \pi) = |g| - \mu(\pi) \cdot \nu^{'}(g, \Lambda)$$

The proof is along similar lines to the proof of Theorem 4.4, and we omit the details here. The main idea is that, ignoring edges close to the boundary, an edge in $N^i(\pi)$ is completely contained in $\nu^{'}(g, \Lambda)$ different queries from $\mathcal{Q}(g, \Lambda)$, and serves to reduce the clustering number of each of those queries by one. The final reduction in the average clustering number depends on the fraction of edges of the SFC that are along each direction, and their relation with the query set $\mathcal{Q}(g, \Lambda)$.

### 7.2. Connected Queries

In this section, we investigate the construction of optimal SFCs for a query class that is broader than the class of rectangular queries. In particular, we consider a natural class of queries that we call *connected queries*.

*Definition* 7.5. A query $g$ is called connected iff the graph $E(g)$ induced by $g$ on $E(U)$ is connected.

The class of connected queries models all those queries where it is possible to go from a point within the query to another point in the query without exiting the query. For instance, the queries shown in Figures 6(b) and 7(b) are both connected queries. Clearly, every rectangular query is also connected.

Theorem 5.2 states that for rectangular queries of a fixed size, there is always a continuous SFC that is optimal. It is interesting to see whether this is true for the case of connected queries. So, we ask two questions here:

(1) For a connected query shape, does there always exist a continuous SFC which is optimal or near-optimal?
(2) For a connected query shape, does this always exist a near-continuous SFC which is optimal, or perhaps near-optimal?

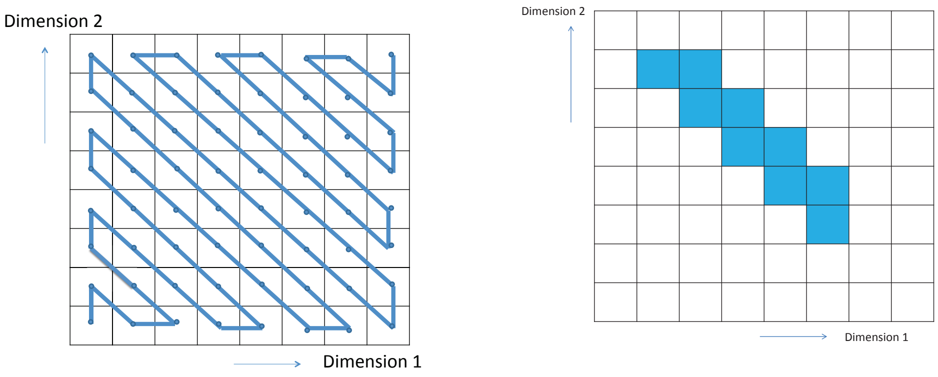
In investigating these questions, we found the following results.

OBSERVATION 7.6 (NEAR-CONTINUOUS VERSUS CONTINUOUS SFCS). *There exist connected query classes where the performance of a near-continuous SFC can be significantly better than that of the best continuous SFC for that query.*

To see this, consider the example in Figure 6. The query $g$ that is shown is connected, and let $k$ be the number of rows occupied by $g$. The near-continuous SFC $\pi_1$ shown has a clustering number equal to 2, regardless of $k$. Let $\pi$ be an arbitrary continuous SFC. Consider the query set formed by all possible translations of $g$. Then we have $\nu(g) = (k-1, k-1)$ and by Theorem 4.2, we get $\lim_{n\to\infty} c(\mathcal{T}(g), \pi) = 2(k-1) + 1 - (k-1) = k$. Therefore we conclude that the performance of any continuous SFC can be $k/2$ times worse than that of a specific near-continuous SFC, for this connected query. By increasing $k$, this gap can be made as large as we like.

OBSERVATION 7.7 (NEAR-CONTINUOUS VERSUS NOT NEAR-CONTINUOUS). *There exist connected query classes where the performance of an SFC that is not near-continuous can be significantly better than that of the best near-continuous SFC for the query.*

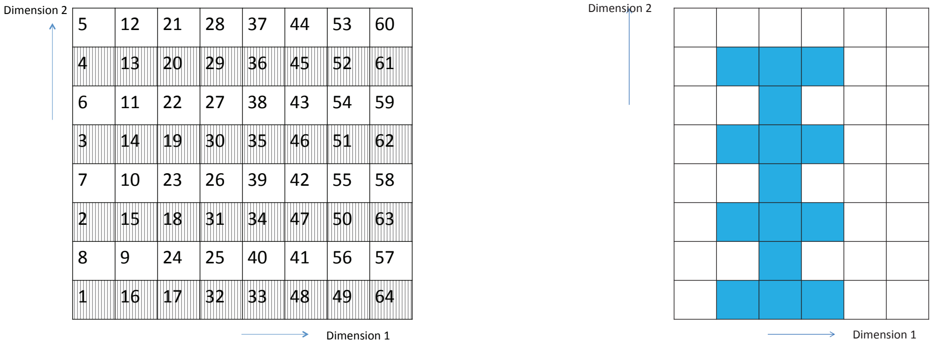
To see this, consider the example in Figure 7, showing a specific connected query $q$ and a non-nearcontinuous SFC $\pi_2$. Consider the query set $\mathcal{T}(q)$. Let $k$ be the number of rows in $q$ which have three cells within $q$. Let $\pi'$ be an arbitrary 2-dimensional near-continuous SFC. Assume $N^i(\pi'), 1 \leq i \leq 4$ be respectively the subset of neighboring pairs $\gamma = (\alpha, \beta) \in N(\pi')$ such that $\alpha - \beta$ equal to $\pm(1,0), \pm(0,1), \pm(1,1), \pm(1,-1)$. According to the Definition of $\nu^{'}$, we get $\nu^{'}(q) = (2k, 2(k-1), 2(k-1), 2(k-1))$. Note that



(a) A near-continuous SFC $\pi_1$

(b) An example of a connected query $g$

Fig. 6. The performance of the near-continuous SFC shown can dominate the performance of any continuous SFC for the above query $g$.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 12 | 21 | 28 | 37 | 44 | 53 | 60 |
| 4 | 13 | 20 | 29 | 36 | 45 | 52 | 61 |
| 6 | 11 | 22 | 27 | 38 | 43 | 54 | 59 |
| 3 | 14 | 19 | 30 | 35 | 46 | 51 | 62 |
| 7 | 10 | 23 | 26 | 39 | 42 | 55 | 58 |
| 2 | 15 | 18 | 31 | 34 | 47 | 50 | 63 |
| 8 | 9 | 24 | 25 | 40 | 41 | 56 | 57 |
| 1 | 16 | 17 | 32 | 33 | 48 | 49 | 64 |

(a) SFC $\pi_2$ that is not near-continuous. The number in a cell indicates the position of the cell in the SFC.

(b) An example of a connected query $q$

Fig. 7. The performance of the SFC shown can dominate the performance of any near-continuous SFC for the above query $q$.

$|q| = 4k - 1$. Thus from Theorem 7.4, we have:

$$\lim_{n\to\infty} c(\mathcal{Q}(q,\Lambda),\pi') \geq 4k - 1 - 2k = 2k - 1$$

Now consider the performance of the SFC $\pi_2$ shown in Figure 7(b). It is clear that $c(q,\pi_2) = 3$ irrespective of $k$, and thus $\lim_{n\to\infty} c(\mathcal{T}(q),\pi_2)$ is 3 regardless of $k$. Therefore we conclude that the performance of any near-continuous SFC is at least $(2k-1)/3$ times as bad as that of a non-nearcontinuous SFC, for the above query class. By increasing $k$, this gap can be made as large as we like.

### 7.3. Query Models

Given a rectangle $r$, and set of all rotations $\Lambda^*$ we note that in Definition 2.7, we have defined $\mathcal{Q}(r,\Lambda^*)$ as the multiset union of the collection of sets $\{\mathcal{T}(\mathcal{P}(g,\lambda))|\lambda \in \Lambda^*\}$.

Suppose we constructed a set of queries $\mathcal{Q}'(r,\Lambda^*)$ not through a multiset union of the above collection of sets, but through a simple union. Then queries that belong to both $\mathcal{T}(\mathcal{P}(g,\lambda_1))$ and $\mathcal{T}(\mathcal{P}(g,\lambda_2))$ for distinct $\lambda_1,\lambda_2 \in \Lambda^*$ are included only once in $\mathcal{Q}'(r,\Lambda^*)$, but multiple times in $\mathcal{Q}(r,\Lambda^*)$. We now show in Lemma 7.9 that the clustering number does not change whether we consider a simple union or a multiset union.

First we prove the following lemma which is used in the proof of Lemma 7.9.

LEMMA 7.8. *For any rectangle $r$ and distinct $\lambda_1, \lambda_2 \in \Lambda'$*

$$|G_{\lambda_1}| = |G_{\lambda_2}|$$

PROOF. Let $r_i, 1 \le i \le d$ be the length of $r$ along dimension $i$. From Lemma 2.8, we know for each $\lambda \in \Lambda^*$, the length of $\mathcal{P}(r,\lambda)$ along dimension $i$ is $r_{\lambda(i)}$.

For two rectangles, the sets formed by all translations of the rectangles are equal if and only if the lengths of the two rectangles along each dimension are equal. In other words, $Q(\lambda_1) = Q(\lambda_2)$ iff for each $i = 1 \ldots d$, $r_{\lambda_1(i)} = r_{\lambda_2(i)}$ So we can rewrite the definition of $G_\lambda$ as $G_\lambda = \{\widehat{\lambda} \in \Lambda^* | r_{\widehat{\lambda}(i)} = r_{\lambda(i)}, \forall 1 \le i \le d\}$.

Assume $\{r_i | 1 \le i \le d\}$ has $K$ distinct numbers. Without loss of generality, we assume $r_i \ne r_j$ for all $1 \le i, j \le K$, and $i \ne j$. For $1 \le i \le K$, let $I_i \subseteq \{1, 2, \ldots d\}$ be the set of numbers such that for each $j \in I_i, r_j = r_i$. It can be seen that for each $\lambda \in \Lambda^*$,

$$|G_\lambda| = \prod_{i=1}^{K} (|I_i|!)$$

That is because for any fixed $\lambda \in \Lambda^*$, $r_{\lambda_1(i)} = r_{\lambda(i)}$ for all $1 \le i \le d$ iff and only of $\lambda_1$ can be equal to $\lambda$ after some permutations in $I_i, 1 \le i \le K$. □

LEMMA 7.9. *For any SFC $\pi$, whether continuous or not:*

$$c(\mathcal{Q}'(r,\Lambda^*),\pi) = c(\mathcal{Q}(r,\Lambda^*),\pi)$$

PROOF. We note that for distinct $\lambda_1, \lambda_2 \in \Lambda^*$, the query sets $\mathcal{T}(r,\lambda_1)$ and $\mathcal{T}(r,\lambda_2)$ are either equal to each other, or completely disjoint from each other. For each $\lambda \in \Lambda^*$, let $Q(\lambda) = \mathcal{T}(\mathcal{P}(r,\lambda))$. Let $\Lambda' \subseteq \Lambda^*$ be the largest subset such that the sets $\{Q(\lambda) | \lambda \in \Lambda'\}$ are all distinct. From the above, we have:

$$c(\mathcal{Q}'(r,\Lambda^*),\pi) = c(\mathcal{Q}(r,\Lambda'),\pi) \tag{16}$$

For each $\lambda \in \Lambda^*$, let $G_\lambda = \{\widehat{\lambda} \in \Lambda^* | Q(\widehat{\lambda}) = Q(\lambda)\}$. The main tool for us here is Lemma 7.8.

From the definition, it follows

$$c(\mathcal{Q}(r,\Lambda^*),\pi) = \frac{\sum_{q \in \mathcal{Q}(r,\Lambda^*)} c(q,\pi)}{|\mathcal{Q}(r,\Lambda^*)|}$$

Using Lemma 7.8, we get $|\mathcal{Q}(r,\Lambda^*)| = |\mathcal{Q}(r,\Lambda')||G_\lambda|$, for some $\lambda \in \Lambda'$. Also, using Lemma 7.8 the numerator of the above reduces to $|G_\lambda| \sum_{q \in \mathcal{Q}(r,\Lambda')} c(q,\pi)$.

$$\begin{aligned} c(\mathcal{Q}(r,\Lambda^*),\pi) &= \frac{|G_\lambda| \sum_{q \in \mathcal{Q}(r,\Lambda')} c(q,\pi)}{|\mathcal{Q}(r,\Lambda')||G_\lambda|} = \frac{\sum_{q \in \mathcal{Q}(r,\Lambda')} c(q,\pi)}{|\mathcal{Q}(r,\Lambda')|} \\ &= c(\mathcal{Q}(r,\Lambda'),\pi) = c(\mathcal{Q}'(r,\Lambda^*),\pi) \quad \text{using Equation 16} \end{aligned}$$

□

## 8. CONCLUSIONS

As observed in [Gutman 1999], when multidimensional data is indexed according to an SFC, the speed of retrieval for a multidimensional query depends on the number and complexity of the one dimensional queries arising as a result. Our analysis provides a lower bound on the average number of one dimensional queries resulting from a given multidimensional shape for any space filling curve, and provides specific space filling curves that meet this lower bound.

We presented results that characterize the clustering properties of space filling curves over query sets that are formed by translations and rotations of a basic query shape. When the basic shape is a rectangle of a fixed size, our analysis presents a near-complete picture in the sense that we obtain matching upper and lower bounds on the clustering number.

One consequence of our work is that any continuous SFC is optimal for rectangular queries of a fixed size, when all rotations are considered. This shows that while the Hilbert curve works well for such queries, since it is a continuous SFC, there is nothing that sets the Hilbert curve apart from say, the row-major curve, when we consider the clustering number as defined here. In fact, when only a subset of rotations are considered for a rectangular query that is not a cube, the optimal SFC, which can be derived from our analysis, may be strictly better than the Hilbert curve. However, we note that for the case when it is allowed to return a superset of the query region, as in the model of Asano *et al.* [Asano et al. 1997], the Hilbert curve has a better performance than the row-major curve, due to its recursive structure.

We also present an analysis of the clustering properties of the popular $d$-dimensional $Z$ SFC. Our general results on continuous SFCs do not hold for the $Z$ SFC, and hence an alternate analysis is necessary for this case.

When the basic query shape is a more general *connected* query, the class of continuous SFCs may not be optimal anymore. We observe that even the more general class of semi-continuous SFCs may not be optimal (or near-optimal) for such queries. An interesting open question is to construct a class of SFCs that can be characterized easily, and that can contain near-optimal or optimal curves for the class of connected queries.

## REFERENCES

ALBER, J. AND NIEDERMEIER, R. 2000. On Multidimensional Curves with Hilbert Property. *Theory Comput. Syst. 33,* 4, 295–312.

ALURU, S. AND SEVILGEN, F. 1997. Parallel domain decomposition and load balancing using space-filling curves. In *Proc. International Conference on High-Performance Computing*. 230 –235.

ASANO, T., RANJAN, D., ROOS, T., WELZL, E., AND WIDMAYER, P. 1997. Space-filling curves and their use in the design of geometric data structures. *Theor. Comput. Sci. 181,* 1, 3–15.

BUGNION, E., ROOS, T., WATTENHOFER, R., AND WIDMAYER, P. 1997. Space Filling Curves versus Random Walks. In *Algorithmic Foundations of Geographic Information Systems*. Springer, 199–211.

FALOUTSOS, C. 1986. Multiattribute hashing using gray codes. *SIGMOD Record 15*, 227–238.

FALOUTSOS, C. 1988. Gray codes for partial match and range queries. *IEEE Trans. Software Engg. 14*, 1381–1393.

GUTMAN, R. 1999. Space-filling curves in geospatial applications. *Dr. Dobb's Journal*. `http://www.drdobbs.com/database/space-filling-curves-in-geospatial-appli/184410998`.

HAVERKORT, H. J. 2011. Recursive tilings and space-filling curves with little fragmentation. *Journal of Computational Geometry 2,* 1, 92–127.

HILBERT, D. 1891. Uber die stetige Abbildung einer Linie auf ein Flachenstuck. *Math. Ann. 38*, 459–460.

JAGADISH, H. V. 1990. Linear clustering of objects with multiple attributes. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. ACM Press, 332–342.

JAGADISH, H. V. 1997. Analysis of the Hilbert curve for representing two-dimensional space. *Information Processing Letters 62*, 17–22.

MATIAS, Y. AND SHAMIR, A. 1987. A video scrambling technique based on space filling curves. In *Proc. Conference on the Theory and Applications of Cryptographic Techniques (CRYPTO)*. Springer, 398–417.

MOON, B., JAGADISH, H. V., FALOUTSOS, C., AND SALTZ, J. H. 2001. Analysis of the clustering properties of the Hilbert space-filling curve. *IEEE Trans. Knowledge and Data Engineering 13,* 1, 124–141.

MORTON, G. 1966. A computer oriented geodetic data base; and a new technique in file sequencing. Tech. rep., IBM.

ORACLE. Oracle spatial and oracle locator. `http://www.oracle.com/technetwork/database/options/spatial/overview/introduction/index.html`.

ORENSTEIN, J. A. AND MERRETT, T. H. 1984. A class of data structures for associative searching. In *Proc. ACM Symposium on Principles of Database Systems (PODS)*. ACM Press, 181–190.

PILKINGTON, J. R. AND BADEN, S. B. 1996. Dynamic partitioning of non-uniform structured workloads with space filling curves. *IEEE Trans. on Parallel and Distributed Systems 7,* 3, 288 – 300.

WARREN, M. AND SALMON, J. 1993. A parallel hashed-octtree N-body algorithm. In *Proc. Supercomputing*. IEEE Computer Society / ACM.

XU, P. AND TIRTHAPURA, S. 2012. On Optimality of Clustering Properties of Space Filling Curves. In *Proc. ACM Symposium on Principles of Database Systems (PODS)*. ACM Press, 215–224.