On the Optimality of Clustering Properties of Space Filling Curves

Pan Xu
Iowa State University
panxu@iastate.edu

Srikanta Tirthapura *
Iowa State University
snt@iastate.edu

ABSTRACT

Space filling curves have for long been used in the design of data structures for multidimensional data. 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 also answer an open problem that was posed by Jagadish in 1997.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Theory

Keywords

space filling curves, clustering, Hilbert curve, lower bound

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 [10] 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. 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 [9].

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 π 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 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.

In an influential work, Moon *et al.* [7] 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 remain unanswered. In particular:

1. **Lower Bound:** Are there any lower bounds on the clustering number of an SFC? This question has been raised before by Jagadish [6] in the context of a 2×2 square query region, but no non-trivial lower bounds were known so far.

2. **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?

3. Are there any general methods for analyzing the clustering number of a curve? Given a query class, which is the best SFC for this class?

*Supported in part by NSF grants 0834743, 0831903.

PODS 2012 May 21–23, 2012, Scottsdale, Arizona, USA.

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 halfplanes, 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.

- **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. This answers a more general version of the question raised by Jagadish [6]. Prior to our work, only upper bounds on the clustering number of specific SFCs, such as the Hilbert SFC were known.

- **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.

  When all possible rotations are considered, surprisingly, *every continuous SFC is optimal for rectangular queries.* The result of Moon *et al.* [7] on the analysis of the Hilbert curve follows as a special case of our result on continuous SFCs.

- **Non-Continuous SFCs.** For the class of SFCs that are not continuous, we show the surprising result that on certain queries a non-continuous SFC may have a much smaller clustering number (i.e. perform much better) than any continuous 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

The work of Moon *et al.* [7] 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 ∞, 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 1 implies the result of [7].

Jagadish [6] 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 2×2 queries, he derived that the average clustering number approaches 2 as n approaches ∞. 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 2×2 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.* [2] present an analysis of the clustering properties of SFCs in two dimensions in a model that is different from ours, and is more "relaxed" in the following respect. 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. In our model the number of clusters is always greater than or equal to the number of clusters in the model of [2]. Alber and Niedermeier [1] present a precise characterization of Hilbert curves in dimensions $d \geq 3$. 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.

Organization of 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 a class of queries, which forms the basis for further analysis. We present the results for a continuous SFC in Section 4, the lower bound on any SFC in Section 5, and we consider non-continuous SFCs and extensions in Section 6.

2. MODEL AND PROBLEM DEFINITION

Let U denote the d dimensional $\sqrt[d]{n} \times \cdots \times \sqrt[d]{n}$ grid of n cells. We assume $\sqrt[d]{n} = 2^k$ for some positive integer k. 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 $\sum_{i=1}^{d} |x_i - y_i|$.

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

Some popularly used space filling curves are the Z-curve [10, 8] (also known as the Morton ordering), the Hilbert curve [5], and the Gray code curve [3, 4]. Figures 1, 2, and 3 show the Hilbert curve, the row-major curve, and the Z curve, respectively.

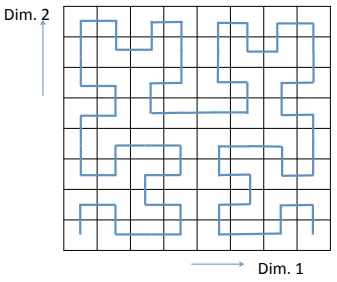


Figure 1: The Hilbert SFC in two dimensions

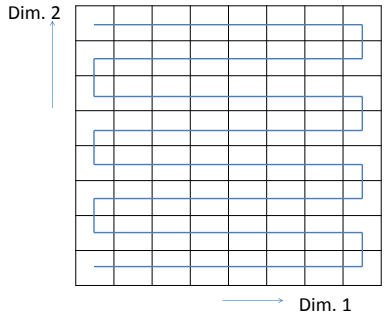


Figure 2: The row-major SFC

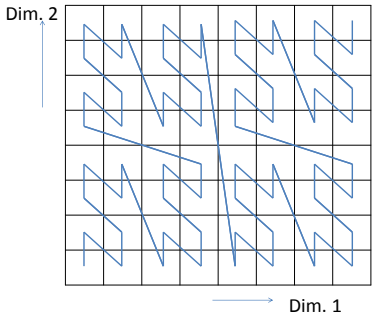


Figure 3: The Z-SFC

DEFINITION 2. *An SFC π 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, the row-major curve (Figure 2) and the Hilbert curve are both continuous SFCs while the Z-curve is not.

DEFINITION 3. *A set of cells $C \subseteq U$ is said to be a "cluster" of an SFC π if the cells of C are numbered consecutively by π.*

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

See Figure 4 for an example of the above definition.

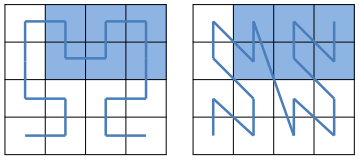


Figure 4: For the same query region shown, the Hilbert curve has a clustering number of 1, and the Z curve has a clustering number of 2.

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 5. *The average clustering number of an SFC π 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. Other definitions of rotation are also possible, and they essentially lead to the same results as we get here. For example, in two dimensions, a 2×3 rectangle can be rotated to a 3×2 rectangle by interchanging dimensions 1 and 2. More precisely, let Λ^* 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 λ. For any $\lambda \in \Lambda^*$, we define the rotation of a cell $x \in U$ under λ 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 λ is defined as:

$$\mathcal{P}(g, \lambda) = \{\mathcal{P}(v, \lambda) | v \in g\}$$

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 δ yields a new query defined as follows (note that "+" denotes vector addition):

$$\mathcal{L}(g, \delta) = \{v + \delta | v \in g\}$$

Given a query g, the set of all possible translations of the query is defined as the set of all possible queries that can be obtained by a translation of g (see Figure 5):

$$\mathcal{T}(g) = \{h \subseteq U | \exists \delta, h = \mathcal{L}(g, \delta)\}$$

DEFINITION 6. *For a query g and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$, the query set $Q(g, \Lambda)$ 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 λ 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 6.2.

For example, suppose $d = 2$ and r is a 2×3 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 2×3 or 3×2 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 2×3 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 1. *Let r be a d-dimensional rectangle and for $1 \leq i \leq d$, let r_i denote the size of r along dimension i. Let $\Lambda \subseteq \Lambda^*$.*

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

Sketch of proof: First, we note that for any $\lambda \in \Lambda$, the set of cells $\mathcal{P}(r, \lambda)$ is still a rectangle, whose side length along dimension i is $r_{\lambda(i)}$. Second, we show that $|\mathcal{T}(r)| = \prod_{i=1}^{d} (\sqrt[d]{n} - r_i + 1)$. To see this, 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 $\prod_{i=1}^{d} (\sqrt[d]{n} - r_i + 1)$. Finally, we note that for any λ, the number of translations of $\mathcal{P}(r, \lambda)$ is the same as the number of translations of r. □

LEMMA 2. *Let g be a query that is not necessarily a rectangle and for $1 \leq i \leq d$, let b_i denote the length of the bounding box $B(g)$ along dimension i. Let $\Lambda \subseteq \Lambda^*$.*

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

Sketch of proof: Recall $|\mathcal{Q}(g, \Lambda)| = \sum_{\lambda \in \Lambda} |\mathcal{T}(\mathcal{P}(g, \lambda))|$ It is possible to show that for any query g, $|\mathcal{T}(g)| = |\mathcal{T}(B(g))|$, so we have $|\mathcal{Q}(g, \Lambda)| = \sum_{\lambda \in \Lambda} |\mathcal{T}(B(\mathcal{P}(g, \lambda)))|$. Next we note that for $\lambda \in \Lambda$, the size of $B(\mathcal{P}(g, \lambda))$ is $b_{\lambda(i)}$ along dimension i, and the rest of this proof proceeds similar to the proof of Lemma 1. □

3. GENERAL TECHNIQUES

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

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

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 π, where each edge goes from a cell numbered i to a cell numbered $(i+1)$, for some $0 \leq i \leq (n-2)$.

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

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

LEMMA 3. *For any query q and any SFC π,*

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

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 π 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 4×4 grid and SFC π as shown in Figure 5(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 that the clustering number is $c(q, \pi) = 3 - 1 = 2$, which can be verified to be correct.

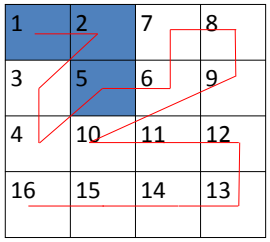
For any query q and a non-empty set of rotations Λ, let query set $Q = \mathcal{Q}(q, \Lambda)$. For a pair of vertices $\alpha, \beta \in U$ (perhaps non-neighboring), let $P_Q(\alpha, \beta)$ be defined as: $P_Q(\alpha, \beta) = \{r \in Q | I(r, \alpha, \beta) = 1\}$.

LEMMA 4.

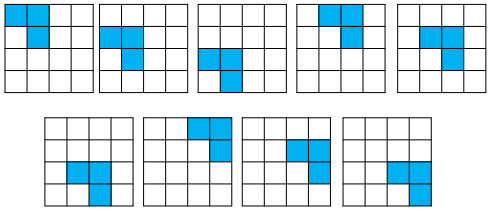
$$c(Q, \pi) = |q| - \frac{\sum_{i=0}^{n-2} |P_Q(\pi^{-1}(i), \pi^{-1}(i+1))|}{|Q|}$$

PROOF. Applying Lemma 3 to Definition 5, we have:

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



(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.

Figure 5: An Example Set of Query Regions for an SFC

□

The above formula relates the clustering number $c(Q, \pi)$ to structural properties of Q and π, 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 π 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 $i = 1 \ldots d$, 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.

For any SFC π, 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 π that connect points in U that are at a Manhattan distance of 1, and lie along dimension i.

4.1 Statement of Results

DEFINITION 7. *For an SFC π, vector $\mu(\pi)$ of length d is defined as: $\mu(\pi) = (\mu_1(\pi), \mu_2(\pi), \ldots, \mu_d(\pi))$, where for $i = 1 \ldots d$*

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

It is assumed that the above limits exist for all the SFCs that we consider.

DEFINITION 8. *Given a query g, vector $\nu(g)$ of length d is defined as: $\nu(g) = (\nu_1(g), \ldots, \nu_d(g))$ 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 1. **Continuous SFC, Translations Only:** *For any continuous SFC π, any query g of fixed size, the average clustering number of π 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 Λ.

DEFINITION 9. *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: $\nu(g, \Lambda) = (\nu_1(g, \Lambda), \ldots, \nu_d(g, \Lambda))$ where for $1 \leq i \leq d$,*

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

THEOREM 2. **Continuous SFC, Translations and Rotations:** *For any continuous SFC π, any query g of a fixed size, the average clustering number of π 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 π_1 shown on the left in Figure 6. Though the picture shows an 8×8 grid, the idea for a $m \times m$ grid is that the SFC goes horizontally (mostly) for the top $5m/8$ rows, and then vertically (mostly) for the bottom $3m/8$ 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

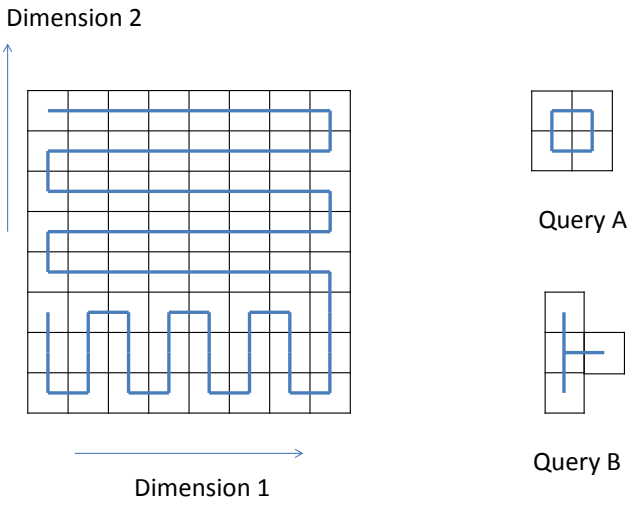


Figure 6: On the left is an SFC π_1 and on the right are two queries A and B.

4 edges, two of them horizontal and two vertical. From Theorem 1, 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 1, 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 1 and 2

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 1 and 2.

From Lemma 4, we have

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

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

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

Since π is continuous, for each $j, 0 \leq j \leq (n-2)$, we have the pair $(\pi^{-1}(j), \pi^{-1}(j+1)) \in N^i(\pi)$ for some $i, 1 \leq i \leq d$. 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:

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

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

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

LEMMA 5. *For any $i, 1 \leq i \leq d$ and any query g,*

$$\rho_{\mathcal{T}(g)}^i \leq \nu_i(g)$$

PROOF. Consider any edge (α, β) 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 (α, β) is parallel to the ith axis, we have the number of translations of g which can include (α, β) 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 $\rho_{\mathcal{T}(g)}^i \leq \nu_i(g)$. □

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 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, 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 (α, β) such that α or β 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 1. We start from Equations 3 and 4. Setting $Q = \mathcal{T}(g)$ in Equation 3,

$$\begin{aligned} S(\mathcal{T}(g), \pi) &= \sum_{i=1}^{d} \sum_{(\alpha,\beta) \in N^i(\pi)} |P_{\mathcal{T}(g)}(\alpha, \beta)| \\ &\leq \sum_{i=1}^{d} |N^i(\pi)| \rho_{\mathcal{T}(g)}^i \quad \text{From Defn. of } \rho \\ &\leq \sum_{i=1}^{d} |N^i(\pi)| \nu_i(g) \quad \text{Using Lemma 5} \end{aligned}$$

Using this back in Equation 1

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

Taking limits on the right side and applying Lemma 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,

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

Taking limits on both sides and applying Lemma 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 2.

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

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

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

From Equation 1, we have:

$$c(\mathcal{Q}(g,\Lambda),\pi) = |g| - \frac{S(\mathcal{Q}(g,\Lambda),\pi)}{|\mathcal{Q}(g,\Lambda)|}$$

$$= |g| - \frac{1}{|\Lambda||Q(\lambda)|}\sum_{\lambda\in\Lambda}\sum_{j=0}^{n-2} |P_{Q(\lambda)}(\pi^{-1}(j),\pi^{-1}(j+1))|$$

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

Applying Theorem 1:

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

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 β is at a Manhattan distance of 1 from some cell α 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 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 α in $G(g)$. The contribution of α 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 3. *For any continuous SFC π, 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 2, we have

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

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

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

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 $i, 1 \leq i \leq d$. $\nu_i(g,\Lambda^*) = \frac{|E(g)|}{d}$

$$\begin{aligned}A &= |g| - \mu(\pi)\left[\frac{|E(g)|}{d}, \frac{|E(g)|}{d}, \ldots, \frac{|E(g)|}{d}\right] \\ &= |g| - \frac{|E(g)|}{d} \text{ since } \textstyle\sum_{i=1}^{d}\mu_i(\pi) = 1 \\ &= \frac{S_g}{2d} \text{ using Lemma 7}\end{aligned}$$

□

Since the Hilbert curve is a continuous curve, the result of Moon *et al.* [7] follows from the above theorem.

4.4 Symmetric SFCs

We say that a continuous SFC π 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 1. *For any symmetric SFC π, 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 2, and then using a similar technique used in the proof of Theorem 3. The difference being that in Theorem 3 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 [7], 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 $\nu^{max} = \nu^{max}(r, \Lambda) = \max_{1 \leq i \leq d} \nu_i(r, \Lambda)$. The main results in this section are stated in Theorems 4 and 5.

THEOREM 4. *Given a rectangular query r of a fixed size and a non-empty set of rotations $\Lambda \subseteq \Lambda^*$, for any SFC π (not necessarily continuous), if $\lim_{n\to\infty} c(\mathcal{Q}(r, \Lambda), \pi)$ exists, then*

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

THEOREM 5. *For a rectangular query r of a fixed size and $\Lambda \subseteq \Lambda^*$, there exists a continuous SFC π 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 π is optimal.

COROLLARY 2. *For a rectangular query r of a fixed size, if all possible rotations are considered, then* any *continuous SFC π is optimal.*

5.2 Proofs of Theorems 4 and 5

To prove Theorem 4, we need the following lemma.

LEMMA 8. *For any SFC π and any pair $(\alpha, \beta) \in N(\pi)$, query set $Q = \mathcal{Q}(r, \Lambda)$, we have:*

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

PROOF. Recall that $P_Q(\alpha, \beta) = \{q \in Q | I(q, \alpha, \beta) = 1\}$. Let $\gamma = \{\alpha, \beta\}$. For each $t \in P_Q(\alpha, \beta)$, we have $\gamma \subseteq t$. Since t is a rectangle it must also be true that the bounding box of γ, $B(\gamma)$ is contained in t. Since $B(\gamma)$ is a rectangle, there is at least one neighbor of α, say α' such that $\alpha' \in B(\gamma)$, and hence $\alpha' \in t$. Note that it is possible $\alpha' = \beta$, if β is at a Manhattan distance of 1 from α.

Let $\gamma' = \{\alpha, \alpha'\}$. Since $\gamma' \subseteq t$, we have $t \in P_Q(\gamma')$. Thus we have that $P_Q(\gamma) \subseteq P_Q(\gamma')$.

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

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 γ' is parallel to the ith axis. For any $\lambda \in \Lambda$, we have the following:

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

The above can be proved using an argument identical to the one used in Lemma 5. In Lemma 5, this was used to bound the size of $P_Q(\alpha'', \beta'')$ where α'' and β'' are neighbors in a continuous SFC, but this exact argument can be used here too since α and α' are at a Manhattan distance of 1.

Combining Equations 7, 8, and 9,

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

□

PROOF OF THEOREM 4. From Lemma 4, we have:

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

Let $r_i, 1 \leq i \leq d$ denote the length of r along dimension i. Applying Lemma 1, 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 8:

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

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. We construct a continuous SFC whose performance meets the above bound.
Let $j = \text{argmax}_{1 \leq i \leq d} \nu_i(r, \lambda)$, so that $\nu_j = \nu^{max}$. Consider SFC S^j defined as follows:

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

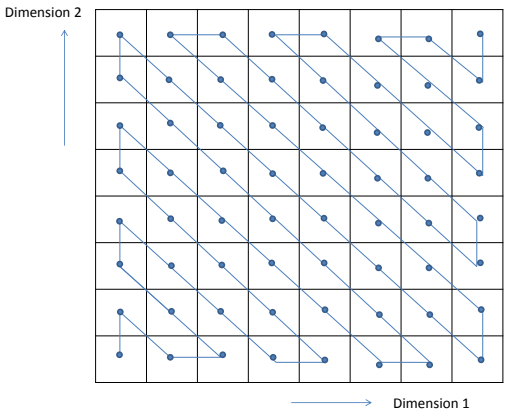
We can check that:

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

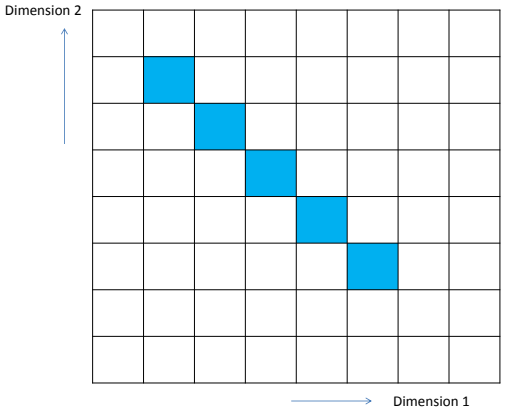
From Theorem 2, we get that for any rectilinear query r and rotation set Λ,

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

So from Theorem 4, we conclude that for any rectangle query r and any nonempty set of rotations $\Lambda \subseteq \Lambda^*$, S^j is optimal among all SFCs. □



(a) A non-continuous SFC π



(b) An example query q

Figure 7: The performance of a non-continuous SFC can dominate the performance of any continuous SFC for the above query.

PROOF OF COROLLARY 2. Consider a continuous SFC π. Using Theorem 3,

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

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 4 is $|r| - \nu^{\max} = |r| - \frac{|E(r)|}{d}$. Proceeding similarly to the proof of Theorem 3, we get the above expression to be $\frac{S_r}{2d}$. Thus, the performance of π meets the lower bound, showing that it is optimal. □

6. EXTENSIONS

6.1 Noncontinuous SFCs

We now consider the performance of SFCs that are not continuous. For rectangular queries of a fixed size, Theorem 5 shows that a non-continuous SFC cannot outperform the best possible continuous SFC. It is natural to ask if the class of continuous SFCs contains an optimal SFC for a general query.

We show that this is not true in general. *There exist query classes where the performance of a non-continuous SFC can be much better than that of the best continuous SFC for that query.* In Figure 7(a), we show a specific noncontinuous SFC π and in Figure 7(b), we show a query q. It is clear that $c(q, \pi) = 1$. It is also clear that $\lim_{n\to\infty} c(\mathcal{T}(q), \pi)$ is 1. Though the picture shows a specific noncontinuous SFC for an 8×8 grid, the same SFC can be extended to a $\sqrt{n} \times \sqrt{n}$ grid for an arbitrary n, and the clustering number for $\mathcal{T}(q)$ is still 1.

However, the clustering number of any continuous SFC for $\mathcal{T}(q)$ must be 5, since no two cells of the query can belong to the same cluster of a continuous SFC. The performance of the best continuous SFC is 5 times as bad as that of a non-continuous SFC. Clearly, by constructing queries that are the same shape with $|q|$ cells, the performance of a continuous SFC can be $|q|$ times as bad as that of a noncontinuous SFC.

6.2 Query Models

Given a rectangle r, and set of rotations Λ^* we note that in Definition 6, 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 $\mathcal{Q}(r, \Lambda^*)$. The following lemma holds.

LEMMA 9. *For any SFC π, whether continuous or not:*

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

PROOF. First, 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:

$$c(\mathcal{Q}'(r, \Lambda^*), \pi) = c(\mathcal{Q}(r, \Lambda'), \pi) \tag{10}$$

For each $\lambda \in \Lambda^*$, let $G_\lambda = \{\lambda_1 \in \Lambda^* | Q(\lambda_1) = Q(\lambda)\}$. The main tool for us here is Lemma 10.

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 10, we get $|\mathcal{Q}(r, \Lambda^*)| = |\mathcal{Q}(r, \Lambda')||G_\lambda|$, for some $\lambda \in \Lambda'$. Also, using Lemma 10 the numerator of the above reduces to $|G_\lambda| \sum_{q \in \mathcal{Q}(r, \Lambda')} c(q, \pi)$.

$$\begin{aligned} c(\mathcal{Q}(r, \Lambda^*), \pi) &= \frac{|G_\lambda| \sum_{q \in \mathcal{Q}(r, \Lambda')} c(q, \pi)}{|\mathcal{Q}(r, \Lambda')||G_\lambda|} \\ &= \frac{\sum_{q \in \mathcal{Q}(r, \Lambda')} c(q, \pi)}{|\mathcal{Q}(r, \Lambda')|} \\ &= c(\mathcal{Q}(r, \Lambda'), \pi) \\ &= c(\mathcal{Q}'(r, \Lambda^*), \pi) \quad \text{using Equation 10} \end{aligned}$$

which yields the desired result. □

LEMMA 10. *For any rectangle r and distinct $\lambda_1, \lambda_2 \in \Lambda'$*

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

PROOF. Let $r_i, 1 \leq i \leq d$ be the length of r along dimension i. From Lemma 1, 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_λ as
$G_\lambda = \{\lambda_1 \in \Lambda^* | r_{\lambda_1(i)} = r_{\lambda(i)}, \forall 1 \leq i \leq d\}$.

Assume $\{r_i | 1 \leq i \leq d\}$ has K distinct numbers. Without loss of generality, we assume $r_i \neq r_j$ for all $1 \leq i, j \leq K$, and $i \neq j$. For $1 \leq i \leq K$, 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 $1 \leq i \leq d$ iff and only of λ_1 can be equal to λ after some permutations in $I_i, 1 \leq i \leq K$. □

7. CONCLUSION

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. 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.

When the basic query shape is a more general rectilinear region, we present an analysis of the clustering number for the class of continuous SFCs. An interesting question is to generalize our results to the case an arbitrary SFC, for a general query shape.

References

[1] J. Alber and R. Niedermeier. On Multidimensional Curves with Hilbert Property. *Theory Comput. Syst.*, 33(4):295–312, 2000.

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

[3] C. Faloutsos. Multiattribute hashing using gray codes. *SIGMOD Record*, 15:227–238, June 1986.

[4] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Trans. Software Engg.*, 14:1381–1393, 1988.

[5] D. Hilbert. Uber die stetige Abbildung einer Linie auf ein Flachenstuck. *Math. Ann.*, 38:459–460, 1891.

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

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

[8] G.M. Morton. A computer oriented geodetic data base; and a new technique in file sequencing. Technical report, IBM, 1966.

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

[10] J. A. Orenstein and T. H. Merrett. A class of data structures for associative searching. In *Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems*, PODS '84, pages 181–190, 1984.