Analysis of the Clustering Properties of the Hilbert Space-Filling Curve

Bongki Moon, H.V. Jagadish, Christos Faloutsos, *Member*, *IEEE*, and Joel H. Saltz, *Member*, *IEEE*

Abstract—Several schemes for the linear mapping of a multidimensional space have been proposed for various applications, such as access methods for spatio-temporal databases and image compression. In these applications, one of the most desired properties from such linear mappings is *clustering*, which means the locality between objects in the multidimensional space being preserved in the linear space. It is widely believed that the Hilbert space-filling curve achieves the best clustering [1], [14]. In this paper, we analyze the clustering property of the Hilbert space-filling curve by deriving closed-form formulas for the number of clusters in a given query region of an arbitrary shape (e.g., polygons and polyhedra). Both the asymptotic solution for the general case and the exact solution for a special case generalize previous work [14]. They agree with the empirical results that the number of clusters depends on the hypersurface area of the query region and not on its hypervolume. We also show that the Hilbert curve achieves better clustering than the z curve. From a practical point of view, the formulas given in this paper provide a simple measure that can be used to predict the required disk access behaviors and, hence, the total access time.

Index Terms—Locality-preserving linear mapping, range queries, multiattribute access methods, data clustering, Hilbert curve, space-filling curves, fractals.

1 Introduction

The design of multidimensional access methods is difficult compared to one-dimensional cases because there is no total ordering that preserves spatial locality. Once a total ordering is found for a given spatial or multidimensional database, one can use any one-dimensional access method (e.g., B^{+}-tree), which may yield good performance for multidimensional queries. An interesting application of the ordering arises in a multidimensional indexing technique proposed by Orenstein [19]. The idea is to develop a single numeric index on a one-dimensional space for each point in a multidimensional space, such that, for any given object, the range of indices, from the smallest index to the largest, includes few points not in the object itself.

Consider a linear traversal or a typical range query for a database where record signatures are mapped with multi-attribute hashing [24] to buckets stored on disk. The linear traversal specifies the order in which the objects are fetched from disk, as well as the number of blocks fetched. The number of nonconsecutive disk accesses will be determined by the order of blocks fetched. Although the order of blocks fetched is not explicitly specified in the range query, it is reasonable to assume that the set of blocks fetched can be rearranged into a number of groups of consecutive blocks by a database server or disk controller mechanism [25]. Since it is more efficient to fetch a set of consecutive disk blocks rather than a randomly scattered set in order to reduce additional seek time, it is desirable that objects close together in a multidimensional attribute space also be close together in the one-dimensional disk space. A good clustering of multidimensional data points on the one-dimensional sequence of disk blocks may also reduce the number of disk accesses that are required for a range query.

In addition to the applications described above, several other applications also benefit from a linear mapping that preserves locality:

1. In traditional databases, a multiattribute data space must be mapped into a one-dimensional disk space to allow efficient handling of partial-match queries [22]; in numerical analysis, large multidimensional arrays [6] have to be stored on disk, which is a linear structure.
2. In image compression, a family of methods use a linear mapping to transform an image into a bit string; subsequently, any standard compression method can be applied [18]. A good clustering of pixels will result in a fewer number of long runs of similar pixel values, thereby improving the compression ratio.
3. In geographic information systems (GIS), run-encoded forms of image representations are ordering-sensitive, as they are based on representations of the image as sets of runs [1].
4. Heuristics in computational geometry problems use a linear mapping. For example, for the traveling salesman problem, the cities are linearly ordered and visited accordingly [2].

- *B. Moon is with the Department of Computer Science, University of Arizona, Tucson, AZ 85721-0077. E-mail: bkmoon@cs.arizona.edu.*
- *H.V. Jagadish is with the EECS Department, University of Michigan, Ann Arbor, MI 48109-2122. E-mail: jag@eecs.umich.edu.*
- *C. Faloutsos is with the Department of Computer Science, Carnegie Mellon Uinversity, Pittsburgh, PA 15213-3891. E-mail: christos@cs.cmu.edu.*
- *J.H. Saltz is with the Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail: saltz@cs.umd.edu.*

Manuscript received 18 Mar. 1996; revised 9 Jan. 1998; accepted 19 Aug. 1999.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 104359.

1041-4347/01/$10.00 © 2001 IEEE

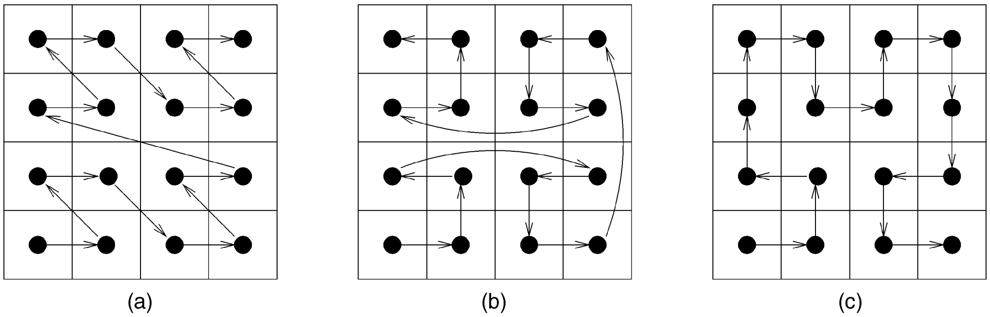


Fig. 1. Illustration of space-filling curves.

5. Locality-preserving mappings are used for bandwidth reduction of digitally sampled signals [4] and for graphics display generation [20].
6. In scientific parallel processing, locality-preserving linearization techniques are widely used for dynamic unstructured mesh partitioning [17].

Sophisticated mapping functions have been proposed in the literature. One based on interleaving bits from the coordinates, which is called z-ordering, was proposed [19]. Its improvement was suggested by Faloutsos [8], using Gray coding on the interleaved bits. A third method, based on the Hilbert curve [13], was proposed for secondary key retrieval [11]. In mathematical context, these three mapping functions are based on different space-filling curves: the *z curve*, the *Gray-coded curve*, and the *Hilbert curve*, respectively. Fig. 1 illustrates the linear orderings yielded by the space-filling curves for a 4×4 grid.

It was shown that under most circumstances, the linear mapping based on the Hilbert space-filling curve outperforms the others in preserving locality [14]. In this paper, we provide analytic results of the clustering effects of the Hilbert space-filling curve, focusing on arbitrarily shaped range queries, which require the retrieval of all objects inside a given hyperrectangle or polyhedron in multidimensional space.

For purposes of analysis, we assume a multidimensional space with finite granularity, where each point corresponds to a grid cell. The Hilbert space-filling curve imposes a linear ordering on the grid cells, assigning a single integer value to each cell. Ideally, it is desirable to have mappings that result in fewer disk accesses. The number of disk accesses, however, depends on several factors, such as the capacity of the disk pages, the splitting algorithm, the insertion order, etc. Here, we use the average number of *clusters*, or *continuous runs*, of grid points within a subspace representing a query region as the measure of the clustering performance of the Hilbert curve. If each grid point is mapped to one disk block, this measure exactly corresponds to the number of nonconsecutive disk accesses, which involve additional seek time. This measure is also highly correlated to the number of disk blocks accessed since (with many grid points in a disk block) consecutive points are likely to be in the same block, while points across a discontinuity are likely to be in different blocks. This measure is used only to render the analysis tractable and some weaknesses of this measure were discussed in [14].

Definition 1.1. *Given a* d*-dimensional query, a cluster is defined to be a group of grid points inside the query that are consecutively connected by a mapping (or a curve).*

For example, there are two clusters in the z curve (Fig. 2a), but only one in the Hilbert curve (Fig. 2b) for the same two-dimensional rectangular query $S_x \times S_y$. Now, the problem we will investigate is formulated as follows:

Given a d-dimensional rectilinear polyhedron representing a query, find the average number of clusters inside the polyhedron for the Hilbert curve.

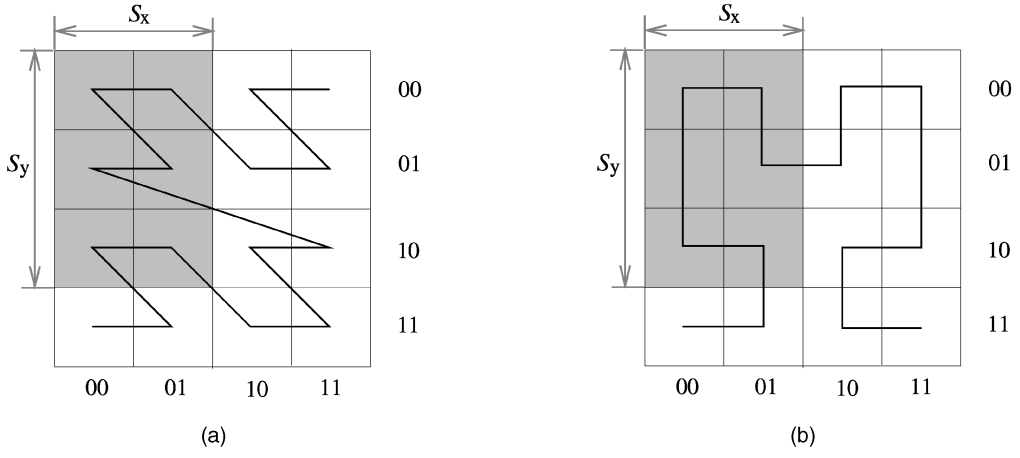
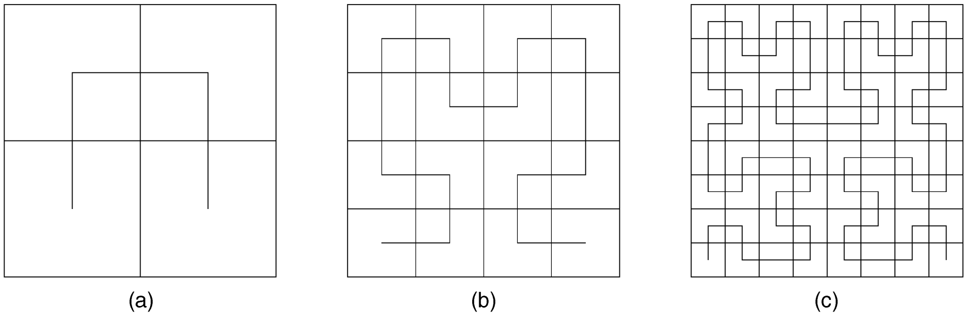


Fig. 2. Illustration of (a) two clusters for the z curve and (b) one cluster for the Hilbert curve.


(a)

(b)

(c)

Fig. 3. The first three steps of the Hilbert space-filling curve: (a) first step, (b) second step, and (c) third step.

The definition of the d-dimensional rectilinear polyhedron is given in Section 3. Note that in the d-dimensional space with finite granularity, for any d-dimensional object, such as spheres, ellipsoids, quadric cones, etc., there exists a corresponding (rectilinear) polyhedron that contains exactly the same set of grid points inside the given object. Thus, the solution to the problem above will cover more general cases concerning any simple connected object of arbitrary shape.

The rest of the paper is organized as follows: Section 2 surveys historical work on space-filling curves and other related analytic studies. Section 3 presents an asymptotic formula of the average number of clusters for d-dimensional range queries of arbitrary shape. Section 4 derives a closed-form exact formula of the average number of clusters in a two-dimensional space. In Section 5, we provide empirical evidence to demonstrate the correctness of the analytic results for various query shapes. Finally, in Section 6, we discuss the contributions of this paper and suggest future work.

2 Historical Survey and Related Work

Peano, in 1890, discovered the existence of a continuous curve which passes through every point of a closed square [21]. According to Jordan's precise notion (in 1887) of continuous curves, Peano's curve is a continuous mapping of the closed unit interval $I = [0, 1]$ into the closed unit square $S = [0, 1]^2$. Curves of this type have come to be called *Peano curves* or *space-filling curves* [28]. Formally,

Definition 2.1. *If a mapping $f : I \rightarrow \mathbf{E}^n (n \geq 2)$ is continuous, and $f(I)$, the image of I under f, has positive Jordan content (area for $n = 2$ and volume for $n = 3$), then $f(I)$ is called a space-filling curve. $\mathbf{E}^n$ denotes an* n*-dimensional Euclidean space.*

Although Peano discovered the first space-filling curve, it was Hilbert, in 1891, who was the first to recognize a general geometric procedure that allows the construction of an entire class of space-filling curves [13]. If the interval I can be mapped continuously onto the square S, then after partitioning I into four congruent subintervals and S into four congruent subsquares, each subinterval can be mapped continuously onto one of the subsquares. If this is carried on ad infinitum, I and S are partitioned into 2^{2n} congruent replicas for $n = 1, 2, 3, \cdots, \infty$. Hilbert demonstrated that the subsquares can be arranged so that the inclusion relationships are preserved, that is, if a square corresponds to an interval, then its subsquares correspond to the subintervals of that interval. Fig. 3 describes how this process is to be carried out for the first three steps. It has been shown that the Hilbert curve is a continuous, surjective, and nowhere differentiable mapping [26]. However, Hilbert gave the space-filling curve, in a geometric form only, for mapping I into S (i.e., two-dimensional Euclidean space). The generation of a three-dimensional Hilbert curve was described in [14], [26]. A generalization of the Hilbert curve, in an analytic form, for higher-dimensional spaces was given in [5].

In this paper, a d-dimensional Euclidean space with finite granularity is assumed. Thus, we use *the kth order approximation* of a d-dimensional Hilbert space-filling curve ($k \geq 1$ and $d \geq 2$), which maps an integer set $[0, 2^{kd} - 1]$ into a d-dimensional integer space $[0, 2^k - 1]^d$.

Notation 2.1. *For $k \geq 1$ and $d \geq 2$, let $\mathcal{H}_k^d$ denote the kth order approximation of a* d*-dimensional Hilbert space-filling curve, which maps $[0, 2^{kd} - 1]$ into $[0, 2^k - 1]^d$.*

The drawings of the first, second, and third steps of the Hilbert curve in Fig. 3 correspond to $\mathcal{H}_1^2$, $\mathcal{H}_2^2$, and $\mathcal{H}_3^2$, respectively.

Jagadish [14] compared the clustering properties of several space-filling curves by considering only 2 × 2 range queries. Among the z curve (2.625), the Gray-coded curve (2.5), and the Hilbert curve (2), the Hilbert curve was the best in minimizing the number of clusters. The numbers within the parentheses are the average number of clusters for 2 × 2 range queries. Rong and Faloutsos [23] derived a closed-form expression of the average number of clusters for the z curve, which gives 2.625 for 2 × 2 range queries (exactly the same as the result given in [14]) and, in general, approaches one-third of the perimeter of the query rectangle plus two-thirds of the side length of the rectangle in the unfavored direction. Jagadish [16] derived closed-form, exact expressions of the average number of clusters for the Hilbert curve in a two-dimensional grid, but only for 2 × 2 and 3 × 3 square regions. This is a special case of the more general formulae derived in this paper.

Abel and Mark [1] reported empirical studies to explore the relative properties of such mapping functions using various metrics. They reached the conclusion that the Hilbert ordering deserves closer attention as an alternative to the z curve ordering. Bugnion et al. estimated the average number of clusters and the distribution of intercluster intervals for two-dimensional rectangular queries. They derived the estimations based on the fraction of vertical and horizontal edges of any particular space-filling curve.

TABLE 1
Definition of Symbols

Symbol	Definition
d	Number of dimensions
$(x_1, ..., x_d)$	Coordinates of a grid point in a d-dimensional grid space
$\mathcal{H}_k^d$	k-th order approximation of the d-dimensional Hilbert curve
φ_i	Number of *i-oriented* $\mathcal{H}_{k-1}^d$ vertices in a $\mathcal{H}_k^d$
$\varepsilon_{i,k}$	Number of *i-oriented* edges in a *d-oriented* $\mathcal{H}_k^d$
S_i^+	Number of interior grid points which face i^+-*surface*
S_i^-	Number of interior grid points which face i^--*surface*
p_i^+	Probability that the predecessor of a grid point is its i^+-*neighbor*
p_i^-	Probability that the predecessor of a grid point is its i^--*neighbor*
$\mathcal{S}_q$	Total surface area of a given d-dimensional rectilinear polyhedral query q
$\mathcal{N}_d$	Average number of clusters within a given d-dimensional rectilinear polyhedron

However, those fractions were provided only for a 2-dimensional space and without any calculation or formal verification. In this paper, we formally prove that, in a d-dimensional space, the d different edge directions approach the uniform distribution, as the order of the Hilbert curve approximation grows into infinity.

Several closely related analyses for the average number of two-dimensional quadtree nodes have been presented in the literature. Dyer [7] presented an analysis for the best, worst, and average case of a square of size $2^n \times 2^n$, giving an approximate formula for the average case. Shaffer [27] gave a closed formula for the exact number of blocks that such a square requires when anchored at a given position (x, y); he also gave a formula for the average number of blocks for such squares (averaged over all possible positions). Some of these formulae were generalized for arbitrary 2-dimensional and d-dimensional rectangles [9], [10].

3 Asymptotic Analysis

In this section, we give an asymptotic formula for the clustering property of the Hilbert space-filling curve for general polyhedra in a d-dimensional space. The symbols used in this section are summarized in Table 1. The polyhedra we consider here are not necessarily convex, but are *rectilinear* in the sense that any *(d-1)*-dimensional polygonal surface is perpendicular to one of the d coordinate axes.

Definition 3.1. *A* rectilinear polyhedron *is bounded by a set V of polygonal surfaces each of which is perpendicular to one of the d coordinate axes, where V is a subset of* $\mathbf{R}^d$ *and homeomorphic*[1] *to a* (d-1)*-dimensional sphere* S^{d-1}.

For $d = 2$, the set V is, by definition, a *Jordan curve* [3], which is essentially a simple closed curve in $\mathbf{R}^2$. The set of surfaces of a polyhedron divides the d-dimensional space $\mathbf{R}^d$ into two connected components, which may be called the *interior* and the *exterior*.

The basic intuition is that each cluster within a given polyhedron corresponds to a segment of the Hilbert curve connecting a group of grid points in the cluster, which has two endpoints adjacent to the surface of the polyhedron. The number of clusters is then equal to half the number of endpoints of the segments bounded by the surface of the polyhedron. In other words,

Remark 3.1. The number of clusters within a given d-dimensional polyhedron is equal to the number of entries (or exits) of the Hilbert curve into (or from) the polyhedron.

Thus, we expect that the number of clusters is approximately proportional to the perimeter or hypersurface area of the d-dimensional polyhedron ($d \geq 2$). With this observation, the task is reduced to finding a constant factor of a linear function.

Our approach to derive the asymptotic solution largely depends on the *self-similar* nature of the Hilbert curve, which stems from the recursive process of the curve expansion. Specifically, we shall show in the following lemmas that the edges of d different orientations are *uniformly distributed* in a d-dimensional Euclidean space. That is, approximately one dth of the edges are aligned to the ith dimensional axis for each i $(1 \leq i \leq d)$. Here, we mean by *edges*, the line segments of the Hilbert curve connecting two neighboring points. The uniform distribution of the edges provides key leverage for deriving the asymptotic solution. To show the uniform distribution, it is important to understand

[1] Two subsets, X and Y, of Euclidean space are called homeomorphic if there exists a continuous bijective mapping, $f : X \rightarrow Y$, with a continuous inverse f^{-1} [12].

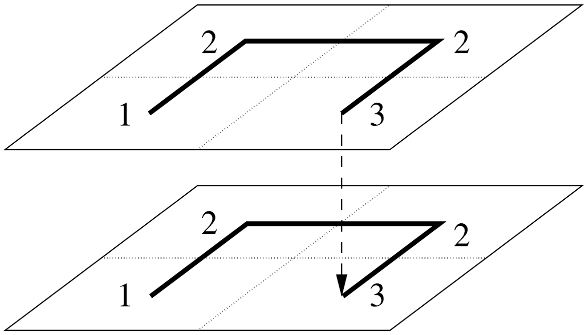


Fig. 4. The 3-dimensional Hilbert curve ($\mathcal{H}_k^3$ with vertices representing $\mathcal{H}_{k-1}^3$ approximations annotated by their orientations.)

- how the kth order approximation of the Hilbert curve is derived from lower order approximations, and
- how the d-dimensional Hilbert curve is extended from the two-dimensional Hilbert curve, which was described only in a geometric form in [13]. (Analytic forms for the d-dimensional Hilbert curves were presented in [5].)

In a d-dimensional space, the kth order approximation of the d-dimensional Hilbert curve $\mathcal{H}_k^d$ is derived from the 1st order approximation of the d-dimensional Hilbert curve $\mathcal{H}_1^d$ by replacing each vertex in the $\mathcal{H}_1^d$ by $\mathcal{H}_{k-1}^d$, which may be rotated about a coordinate axis and/or reflected about a hyperplane perpendicular to a coordinate axis. Since there are 2^d vertices in the $\mathcal{H}_1^d$, the $\mathcal{H}_k^d$ is considered to be composed of 2^d $\mathcal{H}_{k-1}^d$ vertices and (2^d-1) edges, each connecting two of them.

Before describing the extension for the d-dimensional Hilbert curve, we define the *orientations* of $\mathcal{H}_k^d$. Consider $\mathcal{H}_1^d$, which consists of 2^d vertices and (2^d-1) edges. No matter where the Hilbert curve starts its traversal, the coordinates of the start and end vertices of the $\mathcal{H}_1^d$ differ only in one dimension, meaning that both vertices lie on a line parallel to one of the d coordinate axes. We say that $\mathcal{H}_1^d$ is *i-oriented* if its start and end vertices lie on a line parallel to the ith coordinate axis. For any k $(k>1)$, the orientation of $\mathcal{H}_k^d$ is equal to that of $\mathcal{H}_1^d$ from which it is derived.

Figs. 4 and 5 illustrate the processes that generate $\mathcal{H}_k^3$ from $\mathcal{H}_k^2$ and $\mathcal{H}_k^4$ from $\mathcal{H}_k^3$, respectively. In general, when the dth dimension is added to the *(d-1)*-dimensional Hilbert curve, each vertex of $\mathcal{H}_1^{d-1}$ (i.e, $\mathcal{H}_{k-1}^{d-1}$) is replaced by $\mathcal{H}_{k-1}^d$ of the same orientation except in the 2^{d-1}th one (i.e., the end vertex of $\mathcal{H}_1^{d-1}$), whose orientation is changed from 1*-oriented* to *d-oriented* parallel to the dth dimensional axis. For example, in Fig. 5, the orientations of the two vertices connected by a dotted line have been changed from 1 to 4. Since the orientations of all the other $(2^{d-1}-1)$ $\mathcal{H}_{k-1}^d$ vertices remain unchanged, they are all *j-oriented* for some $j(1 \leq j < d)$. The whole 2^{d-1} $\mathcal{H}_{k-1}^d$ vertices are then replicated by reflection and, finally, the two replicas are connected by an edge parallel to the dth coordinate axis (called *d-oriented* edge) to form a *d-oriented* $\mathcal{H}_k^d$. *In short, whenever a dimension (say, the dth dimension) is added, two d-oriented $\mathcal{H}_{k-1}^d$ vertices are introduced, the number of* 1*-oriented $\mathcal{H}_{k-1}^d$ vertices remains unchanged as two, and the number of $\mathcal{H}_{k-1}^d$ vertices of the other orientations are doubled.*

Notation 3.1. *Let φ_i be the number of i-oriented $\mathcal{H}_{k-1}^d$ vertices in a given d-oriented $\mathcal{H}_k^d$.*

Lemma 1. *For a d-oriented $\mathcal{H}_k^d$ $(d \geq 2)$,*

$$\varphi_i = \begin{cases} 2 & \text{if } i = 1, \\ 2^{d+1-i} & \text{if } 1 < i \leq d. \end{cases} \qquad (1)$$

Proof. By induction on d. □

The following lemma shows that the edges of d different orientations approach the uniform distribution as the order of the Hilbert curve approximation grows into infinity.

Notation 3.2. *Let $\varepsilon_{i,k}$ denote the number of i-oriented edges in a d-oriented $\mathcal{H}_k^d$.*

Lemma 2. *In a* d*-dimensional space, for any i and $j(1 \leq i, j \leq d)$, $\varepsilon_{i,k}/\varepsilon_{j,k}$ approaches unity as k grows to infinity.*

Proof. We begin by deriving recurrence relations among the terms $\varepsilon_{i,k}$ and φ_i. As we mentioned previously, the fundamental operations involved in expanding the Hilbert curve (i.e., from $\mathcal{H}_{k-1}^d$ to $\mathcal{H}_k^d$) are *rotation* and *reflection*. During the expansion of $\mathcal{H}_k^d$, the orientation of a $\mathcal{H}_{k-1}^d$ vertex in a quantized subregion is changed only by rotation; a set of subregions of an orientation are replicated from one of the same orientation, which leaves the directions of their edges unchanged. Consequently,

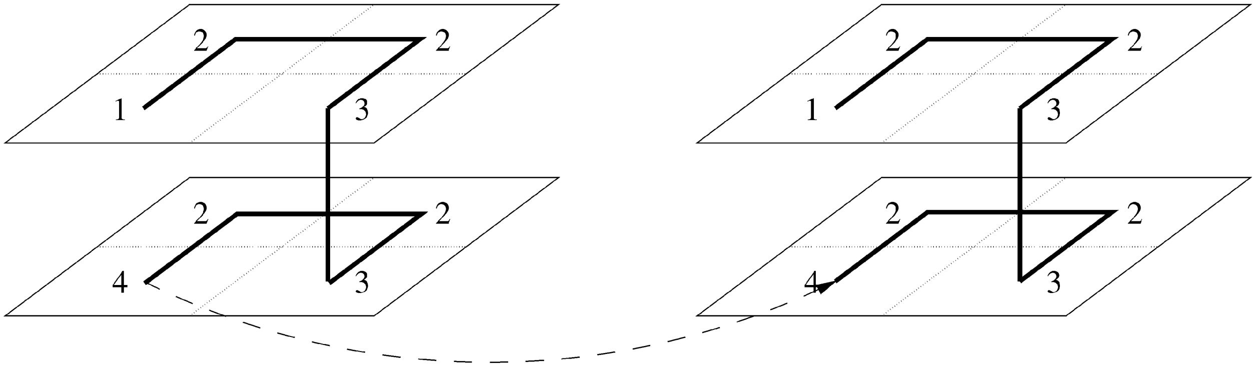


Fig. 5. The 4-dimensional Hilbert curve ($\mathcal{H}_k^4$ with vertices representing $\mathcal{H}_{k-1}^4$ approximations annotated by their orientations.)

any two distinct $\mathcal{H}_{k-1}^d$ vertices of the same orientation contain the same number of edges $\varepsilon_{i,k-1}$ for each direction $i (1 \leq i \leq d)$. Therefore, the set of the 1-*oriented* edges in the $\mathcal{H}_k^d$ consists of 2^{d-1} connection edges (in $\mathcal{H}_1^d$), d-*oriented* edges of the 1-*oriented* $\mathcal{H}_{k-1}^d$ vertices, $(d-1)$-*oriented* edges of the 2-*oriented* $\mathcal{H}_{k-1}^d$ vertices, $(d-2)$-*oriented* edges of the 3-*oriented* $\mathcal{H}_{k-1}^d$ vertices and so on.

By applying the same procedure to the other directions, we obtain

$$\begin{aligned}
\varepsilon_{1,k} &= \varphi_1\varepsilon_{d,k-1} + \varphi_2\varepsilon_{d-1,k-1} + \cdots + \varphi_d\varepsilon_{1,k-1} + 2^{d-1}\\
\varepsilon_{2,k} &= \varphi_2\varepsilon_{d,k-1} + \varphi_3\varepsilon_{d-1,k-1} + \cdots + \varphi_1\varepsilon_{1,k-1} + 2^{d-2}\\
\varepsilon_{3,k} &= \varphi_3\varepsilon_{d,k-1} + \varphi_4\varepsilon_{d-1,k-1} + \cdots + \varphi_2\varepsilon_{1,k-1} + 2^{d-3}\\
&\vdots\\
\varepsilon_{d,k} &= \varphi_d\varepsilon_{d,k-1} + \varphi_1\varepsilon_{d-1,k-1} + \cdots + \varphi_{d-1}\varepsilon_{1,k-1} + 1.
\end{aligned} \tag{2}$$

The initial values are given by $\varepsilon_{i,1} = 2^{d-i}$ and the values of φ_i are in Lemma 1. With the constants in the last terms being ignored, the recurrence relations are completely symmetric. From the symmetry, it can be shown that for any i and $j (1 \leq i, j \leq d)$,

$$\lim_{k\to\infty} \frac{\varepsilon_{i,k}}{\varepsilon_{j,k}} = 1.$$

The proof is complete. □

Now, we consider a d-dimensional grid space, which is equivalent to a d-dimensional Euclidean integer space. In the d-dimensional grid space, each grid point $y = (x_1, \ldots, x_d)$ has $2d$ neighbors. The coordinates of the neighbors differ from those of y by unity only in one dimension. In other words, the coordinates of the neighbors that lie in a line parallel to the ith axis must be either $(x_1, \ldots, x_i+1, \ldots, x_d)$ or $(x_1, \ldots, x_i-1, \ldots, x_d)$. We call them the i^+-*neighbor* and the i^--*neighbor* of y, respectively.

Butz showed that any unit increment in the Hilbert order produces a unit increment in one of the d coordinates and leaves the other $d-1$ coordinates unchanged [5]. The implication is that, for any grid point y, both neighbors of y, in the linear order imposed by the Hilbert curve, are chosen from the $2d$ neighbors of y in the d-dimensional grid space. Of the two neighbors of y in the Hilbert order, the one closer to the start of the Hilbert traversal is called the *predecessor* of y.

Notation 3.3. *For a grid point y in a* d*-dimensional grid space, let p_i^+ be the probability that the predecessor of y is the i^+-neighbor of y and let p_i^- be the probability that the predecessor of y is the i^--neighbor of y.*

Lemma 3. *In a sufficiently large d-dimensional grid space, for any i $(1 \leq i \leq d)$,*

$$p_i^+ + p_i^- = \frac{1}{d}.$$

Proof. Assume y is a grid point in a d-dimensional space and z is its predecessor. Then, the edge $\overline{yz}$ adjacent to y and z is parallel to one of the d dimensional axes. From Lemma 2 and the recursive definition of the Hilbert curve, the probability that $\overline{yz}$ is parallel to the ith dimensional axis is d^{-1} for any i $(1 \leq i \leq d)$. This implies that the probability that z is either i^+-*neighbor* or i^--*neighbor* of y is d^{-1}. □

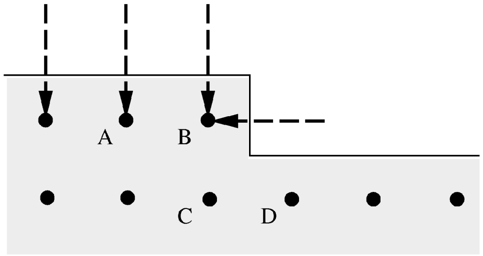


Fig. 6. Illustration of grid points facing surfaces.

For a d-dimensional rectilinear polyhedron representing a query region, the number, sizes, and shapes of the surfaces can be arbitrary. Due to the constraint of surface alignment, however, it is feasible to classify the surfaces of a d-dimensional rectilinear polyhedron into $2d$ different kinds: For any i $(1 \leq i \leq d)$,

- If a point y is inside the polyhedron and its i^+-*neighbor* is outside, then the point y faces an i^+-*surface*.
- If a point y is inside the polyhedron and its i^--*neighbor* is outside, then the point y faces an i^--*surface*.

For example, Fig. 6 illustrates grid points which face surfaces in a two-dimensional grid space. The shaded region represents the inside of the polyhedron. Assuming that the first dimension is vertical and the second dimension is horizontal, grid points A and D face a 1^+-*surface* and grid point B (on the convex) faces both a 1^+-*surface* and a 2^+-*surface*. Although grid point C (on the concave) is close to the boundary, it does not face any surface because all of its neighbors are inside the polyhedron. Consequently, the chance that the Hilbert curve enters the polyhedron through grid point B is approximately twice that of entering through grid point A (or D). The Hilbert curve cannot enter through grid point C.

For any d-dimensional rectilinear polyhedron, it is interesting to see that the aggregate area of i^+-*surface* is exactly as large as that of i^--*surface*. In a d-dimensional grid space, we mean, by surface area, the number of interior grid points that face a given surface of any kind.

Notation 3.4. *For a* d*-dimensional rectilinear polyhedron, let S_i^+ and S_i^- denote the aggregate number of interior grid points that face i^+-surface and i^--surface, respectively.*

Before proving the following theorem, we state an elementary remark without proof.

Remark 3.2. Given a d-dimensional rectilinear polyhedron, $S_i^+ = S_i^-$ for any i $(1 \leq i \leq d)$.

Notation 3.5. *Let $\mathcal{N}_d$ be the average number of clusters within a given* d*-dimensional rectilinear polyhedron.*

Theorem 1. *In a sufficiently large d-dimensional grid space mapped by $\mathcal{H}_k^d$, let S_q be the total surface area of a given rectilinear polyhedral query q. Then,*

$$\lim_{k\to\infty} \mathcal{N}_d = \frac{S_q}{2d}. \tag{3}$$

Proof. Assume a grid point y faces an i^+-*surface* (or an i^--*surface*). Then, the probability that the Hilbert curve enters the polyhedron through y is equivalent to the probability that the predecessor of y is an i^+-*neighbor* (or an i^--*neighbor*) of y. Thus, the expected number of entries through an i^+-*surface* (or an i^--*surface*) is $S_i^+ p_i^+$ (or $S_i^- p_i^-$). Since the number of clusters is equal to the total number of entries into the polyhedron through any of the $2d$ kinds of surfaces (Remark 3.1), it follows that

$$\begin{aligned}\lim_{k\to\infty} \mathcal{N}_d &= \sum_{i=1}^{d}(S_i^+ p_i^+ + S_i^- p_i^-)\\ &= \sum_{i=1}^{d} S_i^+(p_i^+ + p_i^-) \qquad \text{(by Remark 3.2)}\\ &= \sum_{i=1}^{d} S_i^+ \frac{1}{d} \qquad \text{(by Lemma 3)}\\ &= \frac{S_q}{2d}.\end{aligned}$$

The proof is complete. □

Theorem 1 confirms our early conjecture that the number of clusters is approximately proportional to the hypersurface area of a *d*-dimensional polyhedron and provides $(2d)^{-1}$ as the constant factor of the linear function. In a 2-dimensional space, the average number of clusters for the z curve approaches one-third of the perimeter of a query rectangle plus two-thirds of the side length of the rectangle in the unfavored direction [23]. It follows that the Hilbert curve achieves better clustering than the z curve because the average number of clusters for the Hilbert curve is approximately equal to *one-fourth* of the perimeter of a 2-dimensional query rectangle.

Corollary 1. *In a sufficiently large* d*-dimensional grid space mapped by $\mathcal{H}_k^d$, the following properties are satisfied:*

1. *Given an $s_1 \times s_2 \times \cdots \times s_d$ hyperrectangle,*

$$\lim_{k\to\infty} \mathcal{N}_d = \frac{1}{d}\sum_{i=1}^{d}\Big(\frac{1}{s_i}\prod_{j=1}^{d} s_j\Big).$$

2. *Given a hypercube of side length s,*

$$\lim_{k\to\infty} \mathcal{N}_d = s^{d-1}.$$

For a square of side length 2, Corollary 1, item 2 in the list, provides 2 as an average number of clusters, which is exactly the same as the result given in [14].

4 Exact Analysis: A Special Case

Theorem 1 states that as the size of a grid space grows *infinitely*, the average number of clusters approaches half the surface area of a given query region divided by the dimensionality. It does not provide an intuition as to *how rapidly* the number of clusters converges to the asymptotic solution. To address this issue, in this section, we derive a closed-form, exact formula for a two-dimensional *finite* space. We can then measure how closely the asymptotic solution reflects the reality in a finite space by comparing it with the exact formula. Specifically, we assume that a finite $2^{k+n} \times 2^{k+n}$ grid space is mapped by $\mathcal{H}_{k+n}^2$ and a query region is a square of size $2^k \times 2^k$. We first describe our approach and then present the formal derivation of the solution in several lemmas and a theorem. Table 2 summarizes the symbols used in this section.

4.1 Basic Concepts

Remark 3.1 states that the number of clusters within a given query region is equal to the number of entries into the region made by the Hilbert curve traversal. Since each entry is eventually followed by an exit from the region, an entry is equivalent to two cuts of the Hilbert curve by the boundary of the query region. We restate Remark 3.1 as follows:

Remark 4.1. *The number of clusters within a given query region is equal to half the number of edges cut by the boundary of the region.*

Here, we mean by *edges* the line segments of the Hilbert curve connecting two neighboring grid points. Now, we know from Remark 4.1 that deriving the exact formula is reduced to counting the number of edge cuts by the boundary of a $2^k \times 2^k$ query window at all possible positions within a $2^{k+n} \times 2^{k+n}$ grid region. Then, the average number of clusters is simply obtained by dividing this number by twice the number of possible positions of the query window.

Notation 4.1. *Let $\mathcal{N}_2(k, k+n)$ be the average number of clusters inside a $2^k \times 2^k$ square window in a $2^{k+n} \times 2^{k+n}$ grid region.*

The difficulty of counting the edge cuts lies in the fact that, for each edge within the grid region, the number of cuts varies depending on the location of the edge. Intuitively, the edges near the boundary of the grid region are cut less often than those near the center. This is because a smaller number of square windows can cut the edges near the boundary. Thus, to make it easier to count the edge cuts, the grid region $\mathcal{H}_{k+n}^2$ is divided into nine subregions, as shown in Fig. 7. The width of the subregions on the boundary is 2^k. Then, the $2^{k+n} \times 2^{k+n}$ grid region ($\mathcal{H}_{k+n}^2$) can be considered as a collection of 2^{2n} $\mathcal{H}_k^2$ approximations each of which is connected to one or two neighbors by connection edges. From now on, by an *internal edge*, we mean one of the $2^{2k} - 1$ edges in a $\mathcal{H}_k^2$, and by a *connection edge* one that connects two $\mathcal{H}_k^2$ subregions. For example, subregion F includes only one $\mathcal{H}_k^2$ and is connected to subregions B and D by a horizontal and a vertical connection edge, respectively. Subregion B includes $(2^n - 2)$ $\mathcal{H}_k^2$ approximations each of

TABLE 2
Definition of Symbols

Symbol	Definition
t_n	Number of connection edges in the top boundary of a *2^+-oriented* $\mathcal{H}^2_{k+n}$
b_n	Number of connection edges in the bottom boundary of a *2^+-oriented* $\mathcal{H}^2_{k+n}$
s_n	Number of connection edges in the side boundary of a *2^+-oriented* $\mathcal{H}^2_{k+n}$
E_i	A group of edges between grid points
N_i	Number of edge cuts from an edge group E_i
$\psi^{\{R\}}_{i^+,n}$	Number of *i^+-oriented* $\mathcal{H}^2_k$ approximations in the subregion R of a *2^+-oriented* $\mathcal{H}^2_{k+n}$
$\psi^{\{R\}}_{i^-,n}$	Number of *i^--oriented* $\mathcal{H}^2_k$ approximations in the subregion R of a *2^+-oriented* $\mathcal{H}^2_{k+n}$
H_k	Number of horizontal edges in a *2-oriented* $\mathcal{H}^2_k$
V_k	Number of vertical edges in a *2-oriented* $\mathcal{H}^2_k$
$h_k(i)$	Number of horizontal edges in the i-th row from the topmost of a *2^+-oriented* $\mathcal{H}^2_k$
$v_k(i)$	Number of vertical edges in the i-th column from the leftmost of a *2^+-oriented* $\mathcal{H}^2_k$
$\mathcal{N}_2(k, k+n)$	Exact number of clusters covering a $2^k \times 2^k$ square in a $2^{k+n} \times 2^{k+n}$ grid region

which is connected to its two neighbors by connection edges.

Consider an edge (internal or connection) near the center of subregion A,and a horizontal edge in subregion B. An edge in subregion A can be cut by 2^{k+1} square windows, whose positions within the region are mutually distinct. On the other hand, a horizontal edge in subregion B can be cut by a different number of distinct windows, depending on the position of the edge. Specifically, if the edge in subregion B is on the *i*th row from the topmost, then it is cut $2 \times i$ times. The observations we have made are summarized as follows:

1. Every edge (either horizontal or vertical) at least one of whose end points resides in subregion A is cut 2^{k+1} times.
2. Every vertical edge in subregions B and C is cut 2^k times by the top or bottom side of a window.
3. Every horizontal edge in subregions D and E is cut 2^k times by the left or right side of a window.
4. Every connection edge in subregions {B,F,H} is horizontal and resides in the 2^kth row from the topmost and is cut 2^{k+1} times by the left and right sides of a window. Similarly, every connection edge in subregions {C,G,I} is horizontal and resides in the 2^kth row from the topmost and is cut twice by the left and right sides of a window.
5. Every connection edge in subregions {D,F,G} is vertical and resides in the first column from the leftmost and is cut twice by the top and bottom sides of a window. Every connection edge in subregions {E,H,I} is vertical and resides in the first column from the rightmost and is cut twice by the top and bottom sides of a window.
6. Every horizontal edge in the *i*th row from the topmost of subregion B is cut $2 \times i$ times by both the left and right sides of a window and every horizontal edge in the *i*th row from the topmost of subregion *C* is cut $2^{k+1} - 2 \times i + 2$ times by both the left and right sides of a window.
7. Every vertical edge in the *i*th column from the leftmost of subregion D is cut $2 \times i$ times by both the top and bottom sides of a window and every vertical edge in the *i*th column from the leftmost of subregion E is cut $2^{k+1} - 2 \times i + 2$ times by both the top and bottom sides of a window.

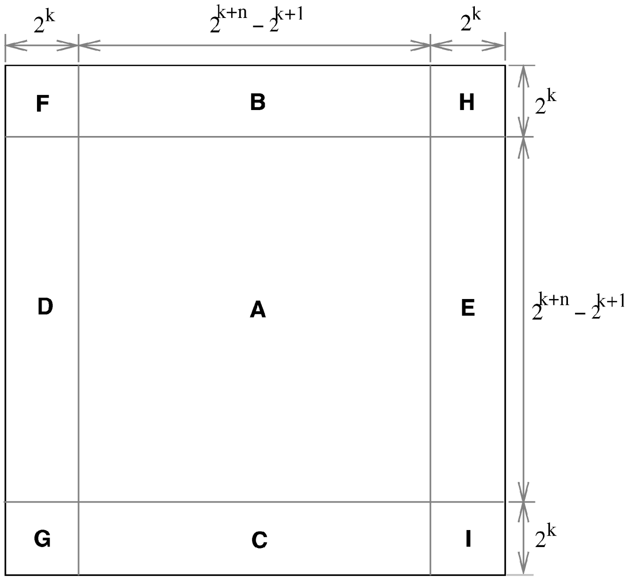


Fig. 7. $\mathcal{H}^2_{k+n}$ divided into nine subregions.

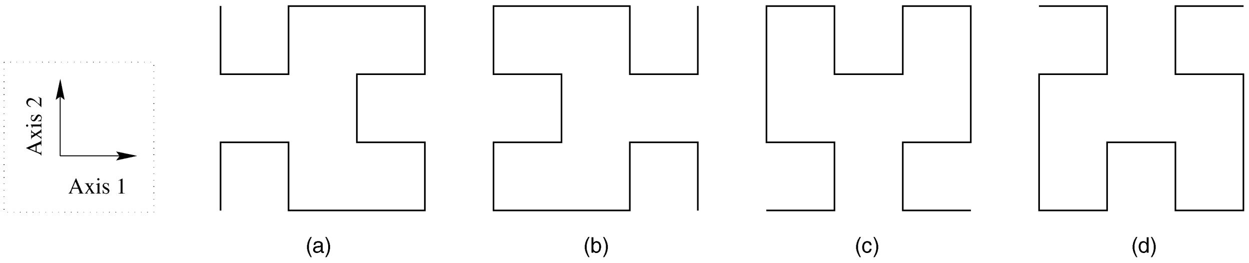


Fig. 8. Four different orientations of $\mathcal{H}_2^2$. (a) 1^+-*oriented*, (b) 1^--*oriented*, (c) 2^+-*oriented*, and (d) 2^--*oriented*.

8. Every horizontal edge in the *i*th row from the topmost of subregions {F,H} is cut i times by either the left or right side of a window.
9. Every horizontal edge in the *i*th row from the topmost of subregions {G,I} is cut $2^k - i + 1$ times by either the left or right side of a window.
10. Every vertical edge in the *i*th column from the leftmost of subregions {F,G} is cut i times by either the top or bottom side of a window.
11. Every vertical edge in the *i*th column from the leftmost of subregions {H,I} is cut $2^k - i + 1$ times by either the top or bottom side of a window.
12. Two connection edges through which the Hilbert curve enters into and leaves from the grid region are cut once each.

From these observations, we can categorize the edges in the $\mathcal{H}_{k+n}^2$ grid region into the following five groups:

1. E_1: a group of edges, as described in observation 1. Each edge is cut 2^{k+1} times.
2. E_2: a group of edges, as described in observations 2 and 3. Each edge is cut 2^k times.
3. E_3: a group of edges, as described in observations 4 and 5. Each connection edge on the top boundary (i.e., subregions {B,F,H}) is cut 2^{k+1} times and any other connection edge is cut twice.
4. E_4: a group of edges, as described in observations 6 and 7. Each edge is cut $2i$ or $2(2^k - i + 1)$ times if it is in the *i*th row (or column) from the topmost (or leftmost).
5. E_5: a group of edges, as described in observations 8 to 11. Each edge is cut i or $2^k - i + 1$ times if it is in the *i*th row (or column) from the topmost (or leftmost).

Notation 4.2. *N_i denotes the number of edge cuts from an edge group E_i.*

In a $\mathcal{H}_{k+n}^2$ grid region, the number of all possible positions of a $2^k \times 2^k$ window is $(2^{k+n} - 2^k + 1)^2$. Since there are two more cuts from observation 12, in addition to $N1, \ldots, N5$, the average number of clusters $\mathcal{N}_2(k, k+n)$ is given by

$$\mathcal{N}_2(k, k+n) = \frac{N_1 + N_2 + N_3 + N_4 + N_5 + 2}{2(2^{k+n} - 2^k + 1)^2}. \tag{4}$$

In the next section, we derive a closed-form expression for each of the edge groups $N_1, \ldots, N_5$.

4.2 Formal Derivation

We adopt the notion of orientations of $\mathcal{H}_k^d$ given in Section 3 and extend it, so that it can be used to derive inductions.

Notation 4.3. *An i-oriented $\mathcal{H}_k^d$ is called i^+-oriented (or i^--oriented) if the ith coordinate of its start point is not greater (or less) than that of any grid point in the $\mathcal{H}_k^d$.*

Fig. 8 illustrates 1^+-*oriented*, 1^--*oriented*, 2^+-*oriented*, and 2^--*oriented* $\mathcal{H}_2^2$ approximations. Note that either of the two end points can be a start point for each curve.

We begin by deriving N_1 and N_3. It appears at the first glance that the derivation of N_1 is simple because each edge in E_1 is cut 2^{k+1} times. However, the derivation of N_1 involves counting the number of connection edges crossing the boundary between subregion A and the other subregions, as well as the number of edges inclusive to subregion A. We accomplish this by counting the number of edges in the complementary set $\overline{E}_1$ (that is, {edges in $\mathcal{H}_{k+n}^2$} $-E_1$). Since $\overline{E}_1$ consists of edges in $4(2^n - 1)$ $\mathcal{H}_k^2$ approximations in boundary subregions B through I and connection edges in E_3, $\mid \overline{E}_1 \mid$ is equal to

$$4(2^n - 1) \times (2^{2k} - 1) + \mid E_3 \mid .$$

To find the number of connection edges in E_3, we define the number of connection edges in different parts of the boundary subregions. In the following, without loss of generality, we assume that the grid region is 2^+-*oriented* $\mathcal{H}_{k+n}^2$.

Notation 4.4. *Let t_n, b_n, and s_n denote the number of connection edges in the top boundary (i.e., subregions {B,F,H}), in the bottom boundary (i.e., subregions {C,G,I}), and in the left or right boundary (i.e., subregions {D,F,G} or {E,H,I}) of a 2^+-oriented $\mathcal{H}_{k+n}^2$, respectively.*

Note that the number of connection edges in subregions {D,F,G} and the number of connection edges in subregions {E,H,I} are identical because the 2^+-*oriented* $\mathcal{H}_{k+n}^2$ is vertically self-symmetric.

Lemma 4. *For any positive integer n,*

$$t_n = 2^{n-1} \quad and \quad b_n + 2s_n = 2(2^n - 1). \tag{5}$$

Proof. Given in Appendix A. □

From Lemma 4, the number of connection edges inclusive to the boundary subregions (i.e., E_3) is given by $t_n + b_n + 2s_n = 5 \times 2^{n-1} - 2$. From this, we can obtain the number of edges in E_1 as well as E_3 and, hence, the number

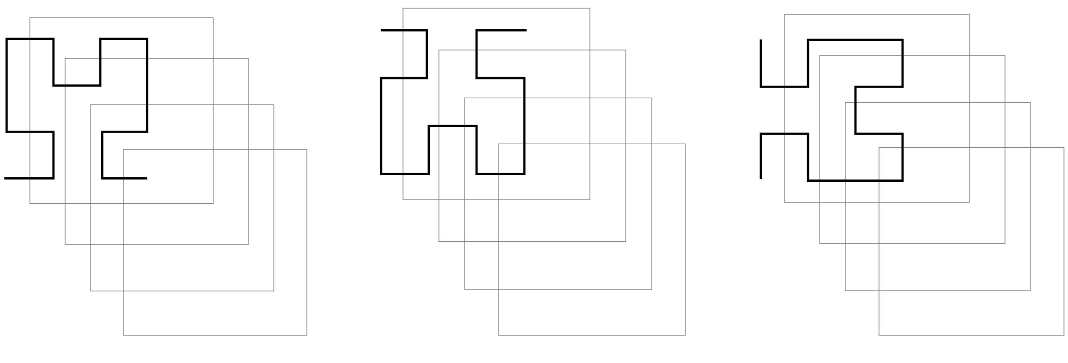


Fig. 9. Three different gradients and cutting windows. (a) *u-gradient*$_2$, (b) *d-gradient*$_2$, and (c) *s-gradient*$_2$.

of cuts from E_1 and E_3. The results are presented in the following lemma.

Lemma 5. *The numbers of edge cuts from E_1 and E_3 are*

$$N_1 = 2(2^n-2)^2 2^{3k} + 3(2^n-2)2^k, \tag{6}$$

$$N_3 = 2^{n+k} + 4(2^n-1). \tag{7}$$

Proof. Given in Appendix A. □

Then, all that we need to derive N_2 is to count the number of vertical edges in subregions {B,C} and the number of horizontal edges in subregions {D,E}. No connection edges in these subregions are involved. Since the number of horizontal (or vertical) edges in a $\mathcal{H}_k^2$ is determined by its orientation, it is necessary to find the number of $\mathcal{H}_k^2$ approximations of different orientations in subregions {B,C,D,E}. In the following, we give notations for the number of horizontal and vertical edges in a $\mathcal{H}_k^2$ and the number of $\mathcal{H}_k^2$ approximations of different orientations in the boundary subregions in Fig. 7.

Notation 4.5. *Let H_k and V_k denote the number of horizontal and vertical edges in a 2-oriented $\mathcal{H}_k^2$, respectively.*

By definition, the numbers of horizontal and vertical edges in a 1-*oriented* $\mathcal{H}_k^2$ are V_k and H_k, respectively.

Notation 4.6. *For a set of subregions $\{R_1, R_2, \ldots, R_j\}$ in Fig. 7, let $\psi_{i^+,n}^{\{R_1,R_2,\ldots,R_j\}}$ and $\psi_{i^-,n}^{\{R_1,R_2,\ldots,R_j\}}$ denote the number of i^+-oriented and i^--oriented $\mathcal{H}_k^2$ approximations in those subregions, respectively.*

Lemma 6. *Given a 2^+-oriented $\mathcal{H}_{k+n}^2$, as depicted in Fig. 7,*

$$\psi_{2^+,n}^{\{B\}} = 2^n - 2, \tag{8}$$

$$\psi_{1^{+,n}}^{\{D\}} + \psi_{1^-,n}^{\{E\}} + \psi_{2^+,n}^{\{C\}} = 2^n - 2, \tag{9}$$

$$\psi_{1^+,n}^{\{C\}} + \psi_{1^-,n}^{\{C\}} + \psi_{2^+,n}^{\{D,E\}} + \psi_{2^-,n}^{\{D,E\}} = 2(2^n-2). \tag{10}$$

Proof. Given in Appendix A. □

From Lemma 6, a closed-form expression of N_2 is derived in the following lemma:

Lemma 7. *The number of edge cuts from E_2 is*

$$N_2 = 2(2^n-2)2^{3k} - 2(2^n-2)2^k. \tag{11}$$

Proof. Given in Appendix A. □

Now, we consider the number of cuts from E_4 and E_5. The edges in these groups are cut a different number of times depending on their relative locations within the $\mathcal{H}_k^2$ which they belong to. Consequently, the expressions for N_4 and N_5 include such terms as $i \times v_k(i)$ and $i \times h_k(i)$. The definitions of $v_k(i)$ and $h_k(i)$ are given below. We call $\mathcal{H}_k^2$ approximations having such terms *gradients*.

Notation 4.7. *Let $h_k(i)$ be the number of horizontal edges in the ith row from the topmost and $v_k(i)$ be the number of vertical edges in the ith column from the leftmost of a 2^+-oriented $\mathcal{H}_k^2$.*

To derive the closed-form expressions for N_4 and N_5, we first define different types of gradients. Consider the 2^+-*oriented* $\mathcal{H}_k^2$ approximations in subregions {B,C,D,E}. From observations 6 and 7, the number of cuts from the horizontal edges in a 2^+-*oriented* $\mathcal{H}_k^2$ in subregion B is $\sum_{i=1}^{2^k} 2ih_k(i)$. Likewise, the number of cuts from the horizontal edges in a 2^+-*oriented* $\mathcal{H}_k^2$ in subregion C is $\sum_{i=1}^{2^k} 2(2^k-i+1)h_k(i)$, and the number of cuts from the vertical edges in a 2^+-*oriented* $\mathcal{H}_k^2$ in subregion D or E is $\sum_{i=1}^{2^k} 2iv_k(i)$. The number of cuts from vertical edges is the same in both subregions D and E because a 2^+-*oriented* $\mathcal{H}_k^2$ is vertically self-symmetric. Based on this, we define three types of gradients for a 2^+-*oriented* $\mathcal{H}_k^2$:

Definition 4.1.

1. *A 2^+-oriented $\mathcal{H}_k^2$ is called u-gradient$_k$ if each of its horizontal edges in the ith row from the topmost is cut i or $2i$ times.*
2. *A 2^+-oriented $\mathcal{H}_k^2$ is called d-gradient$_k$ if each of its horizontal edges in the ith row from the topmost is cut $2^k - i + 1$ or $2(2^k - i + 1)$ times.*
3. *A 2^+-oriented $\mathcal{H}_k^2$ is called s-gradient$_k$ if each of its vertical edges in the ith column from either the leftmost or rightmost is cut i or $2i$ times.*

Fig. 9 illustrates the three different gradients (*u-gradient*$_2$, *d-gradient*$_2$, and *s-gradient*$_2$) and the cutting boundaries of a sliding window. These definitions can be applied to the $\mathcal{H}_k^2$ approximations of different orientations as well, by simply rotating the directions. For example, a 1^+-*oriented* $\mathcal{H}_k^2$ in subregion D is *d-gradient*$_k$ and a 2^--*oriented* $\mathcal{H}_k^2$ in subregion D is *s-gradient*$_k$.

Lemma 8. *Let* $\alpha_k = \sum_{i=1}^{2^k} i h_k(i)$, $\beta_k = \sum_{i=1}^{2^k}(2^k - i + 1)h_k(i)$, *and* $\gamma_k = \sum_{i=1}^{2^k} i v_k(i)$. *Then,*

$$\alpha_k + \beta_k = (2^k + 1)H_k \quad \textit{and} \quad \gamma_k = \frac{1}{2}(2^k + 1)V_k. \qquad (12)$$

Proof. Given in Appendix A. □

Next, we need to know the number of gradients of each type in the boundary subregions B through I so that we can derive N_4 and N_5. For $\mathcal{H}_k^2$ approximations in subregions {B,C,D,E},

- Every 2^+-*oriented* $\mathcal{H}_k^2$ in B is *u-gradient*$_k$.
- Every 2^+-*oriented* $\mathcal{H}_k^2$ in C, 1^+-*oriented* $\mathcal{H}_k^2$ in D, and 1^--*oriented* $\mathcal{H}_k^2$ in E is *d-gradient*$_k$.
- Every 1^+-*oriented* or 1^--*oriented* $\mathcal{H}_k^2$ in C and 2^+-*oriented* or 2^--*oriented* in {D,E} is *s-gradient*$_k$.

The $\mathcal{H}_k^2$ approximations in subregions {F,G,H,I} are dual-type gradients. In other words,

- Each of the 2^+-*oriented* $\mathcal{H}_k^2$ approximations in {F,H} is both *u-gradient*$_k$ and *s-gradient*$_k$.
- The $\mathcal{H}_k^2$ in G is both *d-gradient*$_k$ and *s-gradient*$_k$ because the subgrid is either 2^+-*oriented* or 1^+-*oriented.*
- The $\mathcal{H}_k^2$ in I is both *d-gradient*$_k$ and *s-gradient*$_k$ because the subgrid is either 2^+-*oriented* or 1^--*oriented.*

Thus, in subregions {B,C,D,E}, the number of *u-gradient*$_k$ approximations is $\psi_{2^+,n}^{\{B\}}$, the number of *d-gradient*$_k$ approximations is $\psi_{2^+,n}^{\{C\}} + \psi_{1^+,n}^{\{D\}} + \psi_{1^-,n}^{\{E\}}$, and the number of *s-gradient*$_k$ approximations is

$$\psi_{2^+,n}^{\{D,E\}} + \psi_{2^-,n}^{\{D,E\}} + \psi_{1^-,n}^{\{C\}} + \psi_{1^+,n}^{\{C\}}.$$

In subregions {F,G,H,I}, the number of *u-gradient*$_k$ approximations is two, the number of *d-gradient*$_k$ approximations is two, and the number of *s-gradient*$_k$ approximations is four. From this observation, and Lemma 6 and 8, it follows that

Lemma 9. *The numbers of edge cuts from* E_4 *and* E_5 *are*

$$N_4 = 2(2^n - 2)(2^k + 1)(2^{2k} - 1), \qquad (13)$$

$$N_5 = 2(2^k + 1)(2^{2k} - 1). \qquad (14)$$

Proof. Given in Appendix A. □

Finally, in the following theorem, we present a closed-form expression of the average number of clusters.

Theorem 2. *Given a* $2^{k+n} \times 2^{k+n}$ *grid region, the average number of clusters within a* $2^k \times 2^k$ *query window is*

$$\mathcal{N}_2(k, k+n) = \frac{(2^n - 1)^2 2^{3k} + (2^n - 1)2^{2k} + 2^n}{(2^{k+n} - 2^k + 1)^2}. \qquad (15)$$

Proof. From (4),

$$\mathcal{N}_2(k, k+n) = \frac{(N_1 + N_2 + N_3 + N_4 + N_5 + 2)}{2(2^{k+n} - 2^k + 1)^2} = \frac{((2^n - 1)^2 2^{3k} + (2^n - 1)2^{2k} + 2^n)}{(2^{k+n} - 2^k + 1)^2}.$$

□

For increasing n, $\mathcal{N}_2(k, k+n)$ asymptotically approaches a limit of 2^k, which is the side length of the square query region. This matches the asymptotic solution given in Corollary 1.2 for $d = 2$.

5 Experimental Results

To demonstrate the correctness of the asymptotic and exact analyses presented in the previous sections, we carried out simulation experiments for range queries of various sizes and shapes. The objective of our experiments was to evaluate the accuracy of the formulas given in Theorem 1 and Theorem 2. Specifically, we intended to show that the asymptotic solution is an excellent approximation for general d-dimensional range queries of arbitrary sizes and shapes. We also intended to validate the correctness of the exact solution for a two-dimensional $2^k \times 2^k$ square query.

5.1 Arrangements of Experiments

To obtain exact measurements of the average number of clusters, it was required that we average the number of clusters within a query region at all possible positions in a given grid space. Such exhaustive simulation runs allowed us to validate empirically the correctness of the exact formula given in Theorem 2 for a $2^k \times 2^k$ square query.

However, the number of all possible queries is exponential on the dimensionality. In a d-dimensional $N \times N \times \ldots \times N$ grid space, the total number of distinct positions of a d-dimensional $k \times k \times \ldots \times k$ hypercubic query is $(N - k + 1)^d$. Consequently, for a large grid space and a high dimensionality, each simulation run may require processing an excessively large number of queries, which in turn makes the simulation take too long. Thus, we carried out exhaustive simulations only for relatively small 2-dimensional and three-dimensional grid spaces. Instead, for relatively large or high-dimensional grid spaces, we did statistical simulation by random sampling of queries.

For query shapes, we chose squares, circles, and concave polygons for two-dimensional cases, and cubes, concave polyhedra, and spheres for three-dimensional cases. Fig. 10 illustrates some of the query shapes used in our experiments. In higher dimensional spaces, we used hypercubic and hyperspherical query shapes because it was relatively easy to identify the query regions by simple mathematical formulas.

5.2 Empirical Validation

The first set of experiments was carried out in 2-dimensional grid spaces with two different sizes. The table in Fig. 11a compares the empirical measurements with the exact and asymptotic formulas for a $2^k \times 2^k$ square query. The second column of the table contains the average

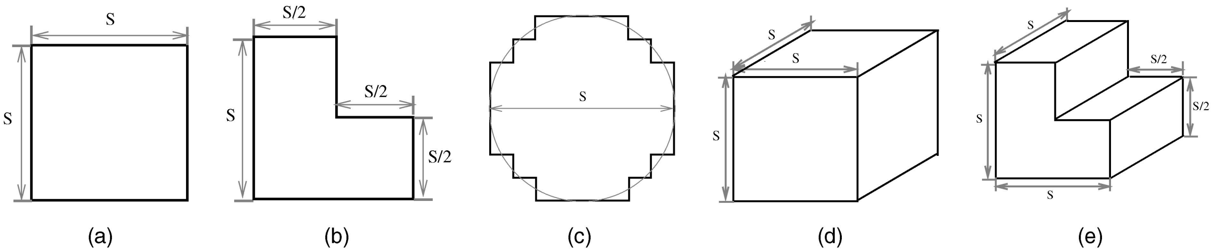
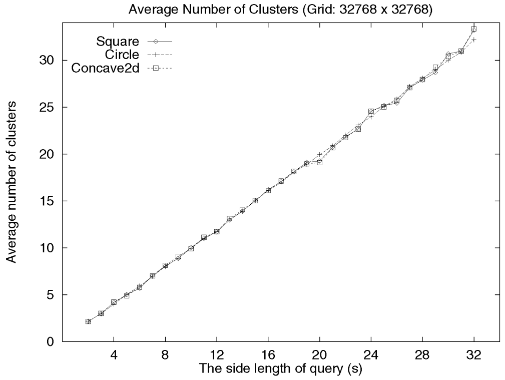


Fig. 10. Illustration of sample query shapes: (a) square, (b) polygon, (c) circle, (d) cube, and (e) polyhedron.

query	empirical	asymptotic	exact
$2^1 \times 2^1$	1.998534	2	2091524/1046529
$2^2 \times 2^2$	3.996328	4	4165936/1042441
$2^3 \times 2^3$	7.992257	8	8266304/1034289
$2^4 \times 2^4$	15.984206	16	16273216/1018081
$2^5 \times 2^5$	31.967807	32	31521824/986049

(a)



(b)

Fig. 11. Average number of clusters for two-dimensional queries: (a) exhaustive simulation (grid: $1,024 \times 1,024$) and (b) statistic simulation (grid: $32K \times 32K$).

numbers of clusters obtained by an exhaustive simulation performed on a $1,024 \times 1,024$ grid space. The numbers in the third and fourth columns were computed by the formulas in Theorem 1 and Theorem 2, respectively. The numbers from the simulation are identical to those from the exact formula ignoring round-off errors. Moreover, by comparing the second and third columns, we can measure how closely the asymptotic formula reflects the reality in a finite grid space.

Fig. 11b compares three different two-dimensional query shapes: squares, circles, and concave polygons. The average number of clusters were obtained by a statistical simulation performed on a $32K \times 32K$ grid space. For the statistical simulation, a total of 200 queries were generated and placed randomly within the grid space for each combination of query shape and size. With a few exceptional cases, the numbers of clusters form a linear curve for each query shape; the linear correlation coefficients are 0.999253 for squares, 0.999936 for circles, and 0.999267 for concave polygons. The numbers are almost identical for the three different query shapes despite their covering different areas. A square covers s^2 grid points, a concave polygon $3s^2/4$ grid points and a circle approximately $\pi s^2/4$ grid points.

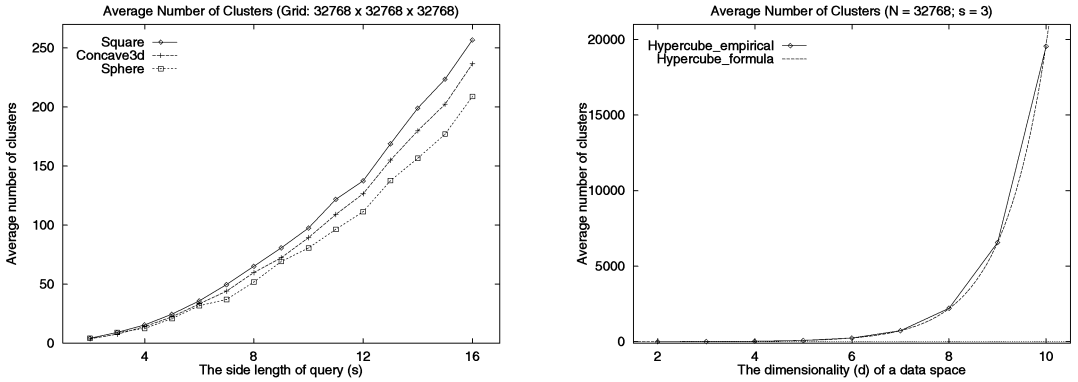
However, this should not be surprising, as the three query shapes have the same length of perimeter for a given side length s. For a circular query of diameter s, we can always find a rectilinear polygon that contains the same set of grid points as the circular query region. And, it is always the case that the perimeter of the rectilinear polygon (as shown in Fig. 10c) is equal to that of a square of side length s. In general, in a two-dimensional grid space, the perimeter of a rectilinear polygon is greater than or equal to that of the minimum bounding rectangle (MBR) of the polygon. This justifies the general approach of using a minimum bounding rectangle to represent a two-dimensional range query because the use of an MBR does not increase the actual number of clusters (i.e., the number of nonconsecutive disk accesses).

A similar set of experiments was carried out in higher dimensional grid spaces. The results in Fig. 12a were obtained by a statistical simulation performed on a $32K \times 32K \times 32K$ grid space. For the statistical simulation, a total of 200 queries were generated and placed randomly within the grid space for each combination of query shape and size. Those in Fig. 12b were obtained by a statistical simulation with 200 random d-dimensional $3 \times 3 \times \ldots \times 3$ hypercubic queries in a d-dimensional $32K \times 32K \times \ldots \times 32K$ grid space ($2 \leq d \leq 10$).

In Fig. 12a, the numbers of clusters form quadratic curves for all the three query shapes, but with slightly different coefficients for the quadratic term. To determine the quadratic functions, we applied the least-square curve fitting method for each query shape. The approximate quadratic functions were obtained as follows:

$$f_{cube}(s) = 1.02307s^2 + 1.60267s + 1.93663$$
$$f_{poly}(s) = 0.947168s^2 + 1.26931s + 1.95395$$
$$f_{sphere}(s) = 0.816674s^2 + 1.27339s + 2.6408.$$

The approximate function $f_{cube}(s)$ for a cubic query confirms the asymptotic solution given in Corollary 1, item 2 in the list, as it is quite close to s^2. Furthermore, Fig. 12b illustrates that the empirical results from the hypercubic queries



(a)

(b)

Fig. 12. Average number of clusters for higher-dimensional queries: (a) three-dimensional queries and (b) *d*-dimensional hypercubic queries.

coincide with the formula (s^{d-1}) even in higher-dimensional spaces.[2] The numbers from the experiments were less than 2 percent off from the formula.

In contrast, the functions $f_{poly}(s)$ and $f_{sphere}(s)$ for concave polyhedral and spherical queries are lower than s^2. The reason is that, unlike in the two-dimensional case, the surface area of a concave polyhedron or a sphere is smaller than that of its minimum bounding cube. For example, the surface area of the polyhedron illustrated in Fig. 10e is $\frac{11}{2}s^2$, while that of the corresponding cube is $6s^2$. For a sphere of diameter $s = 16$, the surface area (i.e., the number of grid points on the surface of the sphere) is 1,248. This is far smaller than the surface area of the corresponding cube, which is 6×16^2. Note that the coefficients of the quadratic terms in $f_{poly}(s)$ and $f_{sphere}(s)$ are fairly close to $\frac{11}{12} = 0.9166\cdots$ and $\frac{1{,}248}{6\times 32^2} = 0.8125$, respectively. This indicates that, in a d-dimensional space ($d \geq 3$), accessing the minimum bounding hyperrectangle of a given query region may incur additional nonconsecutive disk accesses and, hence, supports the argument made in [15] that the minimum bounding rectangle may not be a good approximation of a nonrectangular object.

5.3 Comparison with the Gray-Coded and Z Curves

It may be argued that it is not convincing to make a definitive conclusion that the Hilbert curve is better or worse than others solely on the basis of the average behaviors because the clustering achieved by the Hilbert curve might have a wider deviation from the average than other curves. Therefore, it is desirable to perform a worst-case analysis to determine the bounds on the deviation. A full-fledged worst-case analysis, however, is beyond the scope of this paper. Instead, we measured the worse-case numbers of clusters for the Hilbert curve and compared those for the Gray-coded and z curves in the same simulation experiments.

Fig. 13 and Fig. 14 show the worst-case and average numbers of clusters, respectively. Each figure presents the results from an exhaustive simulation performed on a $1K \times 1K$ two-dimensional space and a statistical simulation performed on a $32K \times 32K \times 32K$ three-dimensional space. The Hilbert curve achieves much better clustering than the other curves in both the worst and average cases. For example, for a two-dimensional square query, the Hilbert curve significantly reduced the numbers of clusters, yielding an improvement of up to 43 percent for the worst-case behaviors and 48 percent for the average cases. For a three-dimensional spherical query, the Hilbert curve achieved an improvement of up to 28 percent from the z curve and 18 percent from the Gray-coded curve for the worst cases and up to 31 percent from the z curve and 22 percent from the Gray-coded curve for the average cases.

Although it is not the focus of this paper, it is worth noting that the Gray-coded curve was not always better than the z curve, which is in contrast to a previous study [14] that the Gray-coded curve achieves better clustering than the z curve for a two-dimensional 2×2 square query. In particular, for two-dimensional circular queries (Fig. 13b and Fig. 14b), the Gray-coded curve was *worse* than the z curve in both the worst and average cases. On the other hand, for two-dimensional square queries, the Gray-coded curve was better than the z curve for the average clustering only by negligible amounts (the two measurements were almost identical, as shown in Fig. 14a). Furthermore, it was surprising that both the Gray-coded and z curves performed exactly the same for the worst-case clustering (the two measurements were completely identical, as shown in Fig. 13a). In a three-dimensional space, however, the Gray-coded curve was clearly better than the z curve for both types of queries in both the worst and average cases.

5.4 Summary

The main conclusions from our experiments are:

- The exact solution given in Theorem 2 matches exactly the experimental results from exhaustive simulations for the square queries of size $2^k \times 2^k$. (See Fig. 11a.)
- The asymptotic solutions given in Theorem 1 and Corollary 1 provide excellent approximations for d-dimensional queries of arbitrary shapes and sizes.

[2] The exponential growth gives rise to the question of whether using the Hilbert curve is a practical technique for clustering high-dimensional data objects. For instance, in a 10-dimensional space, the expected number of clusters was 19,683.

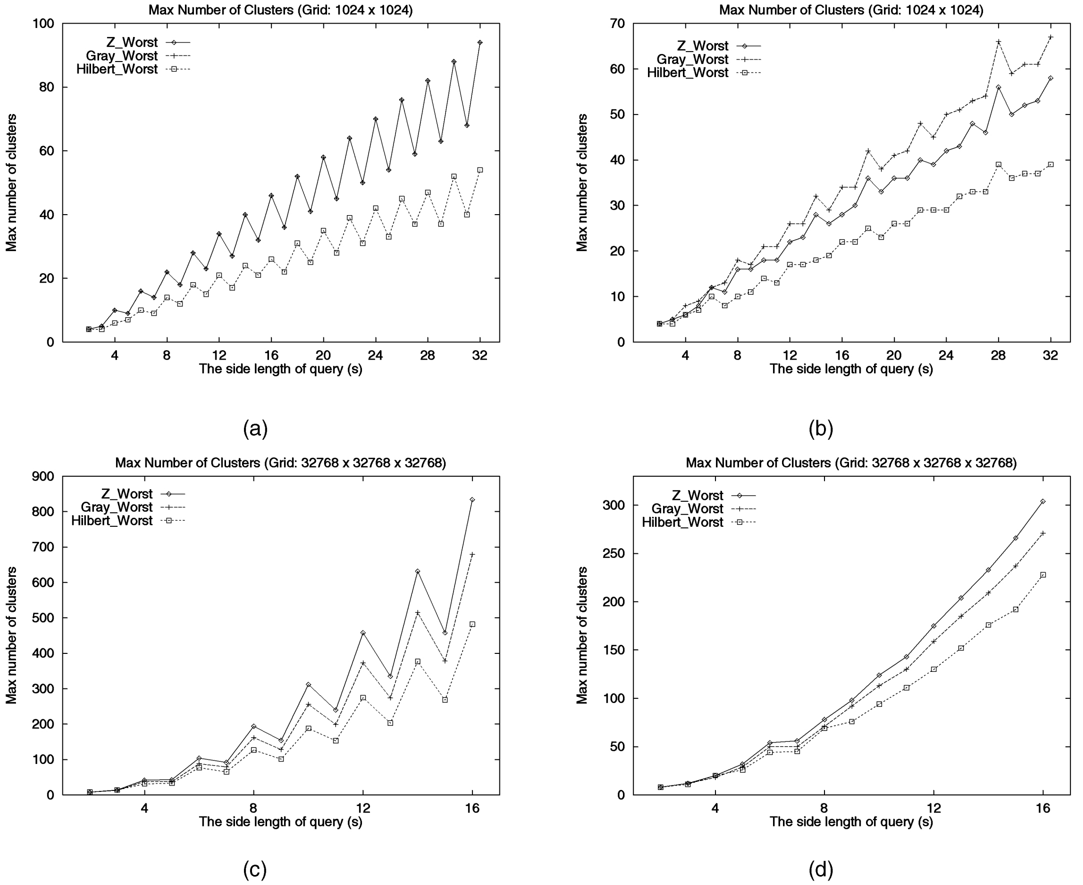


Fig. 13. Worst-case number of clusters for three different space-filling curves: (a) two-dimensional square queries, (b) two-dimensional circular queries, (c) three-dimensional cubic queries, and (d) three-dimensional spherical queries.

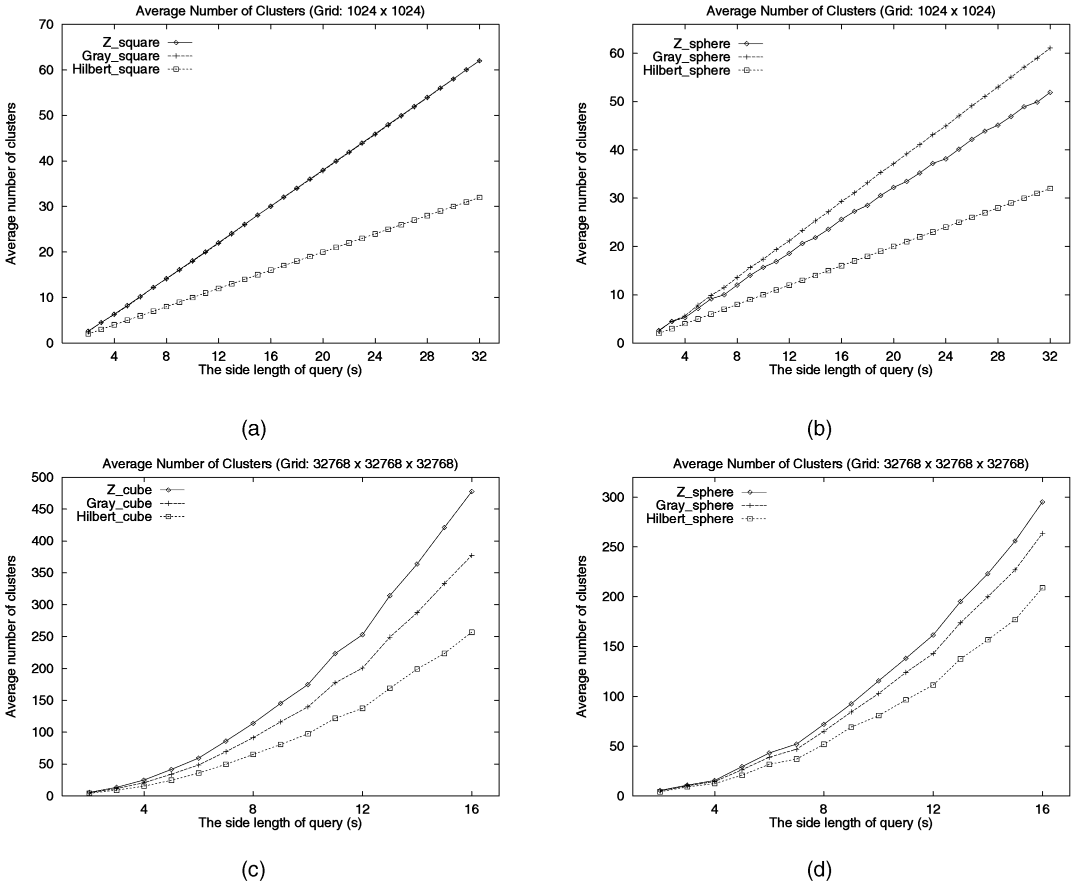
(See Fig. 11b and Fig. 12.) For example, the relative errors did not exceed 2 percent for d-dimensional $(2 \leq d \leq 10)$ hypercubic queries.

- Assuming that blocks are arranged on disk by the Hilbert ordering, accessing the minimum bounding rectangles of a d-dimensional $(d \geq 3)$ query region may increase the number of nonconsecutive accesses, whereas this is not the case for a two-dimensional query.
- The Hilbert curve outperforms the z and Gray-coded curves by a wide margin for both the worst and average case clustering. (See Fig. 13 and Fig. 14.)
- For three-dimensional cubic and spherical queries, the Gray-coded curve outperformed the z curve for both the worst-case and average clustering. However, the clustering by the Gray-coded curve was almost identical to that by the z curve for two-dimensional square queries (in Fig. 13a and Fig. 14a) and clearly worse for two-dimensional circular queries (in Fig. 13b and Fig. 14b).

6 Conclusions

We have studied the clustering property of the Hilbert space-filling curve as a linear mapping of a multidimensional space. Through algebraic analysis, we have provided simple formulas that state the expected number of clusters for a given query region and also validated their correctness through simulation experiments. The main contributions of this paper are:

- Theorem 2 generalizes the previous work done only for a 2×2 query region [14] by providing an exact closed-form formula for $2^k \times 2^k$ square queries for any $k\,(k \geq 1)$. The asymptotic solution given in Theorem 1 further generalizes it for d-dimensional polyhedral query regions $(d \geq 2)$.
- We have proven that the Hilbert curve achieves better clustering than the z curve in a two-dimensional space; the average number of clusters for the Hilbert curve is one-fourth of the perimeter of a query rectangle, while that of the z curve is one-third of the perimeter plus two-thirds of the side length of the rectangle in the unfavored direction [23]. Furthermore, by simulation experiments, we have shown that the Hilbert curve outperforms both the z and Gray-coded curves in two-dimensional and 3-dimensional spaces. We conjecture that this trend will hold even in higher-dimensional spaces.
- We have shown that it may incur extra overhead to access the minimum bounding hyperrectangle for a d-dimensional nonrectangular query $(d \geq 3)$ because it may increase the number of clusters (i.e., nonconsecutive disk accesses).



(a) (b)

(c) (d)

Fig. 14. Average number of clusters for three different space-filling curves: (a) two-dimensional square queries, (b) two-dimensional circular queries, (c) three-dimensional cubic queries, and (d) three-dimensional spherical queries.

The approaches used in this paper can be applied to other space-filling curves. In particular, the basic intuitions summarized in Remark 3.1 and Remark 4.1 are true for any space-filling curves.

From a practical point of view, it is important to predict and minimize the number of clusters because it determines the number of nonconsecutive disk accesses, which, in turn, incur additional seek time. Assuming that blocks are arranged on disk by the Hilbert ordering, we can provide a simple measure that depends only on the perimeter or surface area of a given query region and its dimensionality. The measure can then be used to predict the required disk access behaviors and, thereby, the total access time.

The full-fledged analysis of the worst-case behaviors for the Hilbert curve is left for future research. Future work also includes the extension of the exact analysis for d-dimensional spaces ($d \geq 3$) and the investigation of the distribution of distances between clusters.

Appendix

Proofs

Proof of Lemma 4. A 2^+-*oriented* $\mathcal{H}^2_{k+n}$ approximation is composed of four $\mathcal{H}^2_{k+n-1}$ approximations (two on the top and two on the bottom) and three connection edges. The two $\mathcal{H}^2_{k+n-1}$ approximations on the top half are 2^+-*oriented* and the two $\mathcal{H}^2_{k+n-1}$ approximations on the bottom half are 1^+-*oriented* on the left and 1^+-*oriented* on the right. Among the three edges connecting the four $\mathcal{H}^2_{k+n-1}$ approximations, the horizontal edge is not included in the boundary subregion of the $\mathcal{H}^2_{k+n}$ because the edge resides on the 2^{k+n-1}th row from the topmost of the $\mathcal{H}^2_{k+n}$. The other two vertical connection edges are on the leftmost and rightmost columns and included in the boundary subregion of the $\mathcal{H}^2_{k+n}$. Thus, the main observations are:

1. The number of connection edges in the top boundary subregion of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ is the sum of those in the top boundary subregions of the two 2^+-*oriented* $\mathcal{H}^2_{k+n-1}$ approximations.
2. The number of connection edges in the bottom boundary subregion of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ is the sum of those in the bottom boundary subregions of the 1^+-*oriented* $\mathcal{H}^2_{k+n-1}$ and 1^--*oriented* $\mathcal{H}^2_{k+n-1}$ approximations.
3. The number of connection edges in the left (or right) boundary subregion of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ is the sum of those in the left (or right) boundary subregions of the 2^+-*oriented* $\mathcal{H}^2_{k+n-1}$ and 1^+-*oriented* (or 1^--*oriented*) $\mathcal{H}^2_{k+n-1}$ approximations, plus one for a connection edge.

Since the bottom boundary subregion of a 1^+-*oriented* $\mathcal{H}^2_{k+n-1}$ is equivalent to the right boundary subregion of a 2^+-*oriented* $\mathcal{H}^2_{k+n-1}$, etc., it follows that

$$\begin{aligned} t_n &= 2 \times t_{n-1} \\ b_n &= 2 \times s_{n-1} \\ s_n &= s_{n-1} + b_{n-1} + 1. \end{aligned}$$

Since $t_1 = 1, b_1 = 0$, and $s_1 = 1$, we obtain $t_n = 2^{n-1}$ and

$$b_n + 2s_n = 2(b_{n-1} + 2s_{n-1}) + 2,$$

which yields $b_n + 2s_n = 2(2^n - 1)$. □

Proof of Lemma 5. The $\mathcal{H}^2_{k+n}$ and $\mathcal{H}^2_k$ approximations contain $2^{2(k+n)} - 1$ and $2^{2k} - 1$ edges, respectively. Since there are a total of $4(2^n - 1)$ $\mathcal{H}^2_k$ approximations in the boundary subregions, the total number of edges in E_1 is given by

$$\begin{aligned} &(2^{2(k+n)} - 1) - 4(2^n - 1)(2^{2k} - 1) - (5 \times 2^{n-1} - 2) \\ &\quad = 2^{2k}(2^n - 2)^2 + 3(2^{n-1} - 1). \end{aligned}$$

Because each edge in E_1 is cut 2^{k+1} times, it follows that

$$\begin{aligned} N_1 &= 2^{k+1}(2^{2k}(2^n - 2)^2 + 3(2^{n-1} - 1)) \\ &= 2(2^n - 2)^2 2^{3k} + 3(2^n - 2)2^k. \end{aligned}$$

Among the $5 \times 2^{n-1} - 2$ edges in E_3, t_n edges are cut 2^{k+1} times, and the other $b_n + 2s_n$ edges are cut twice. Therefore,

$$N_3 = 2^{k+1} t_n + 2(b_n + 2s_n) = 2^{n+k} + 4(2^n - 1).$$

□

Proof of Lemma 6. Consider a 2^+-*oriented* $\mathcal{H}^2_{k+n}$, which is composed of four $\mathcal{H}^2_{k+n-1}$ approximations and three connection edges. The number of 2^+-*oriented* $\mathcal{H}^2_k$ approximations in the top subregions (i.e., {B,F,H}) of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ is twice the number of 2^+-*oriented* $\mathcal{H}^2_k$ approximations in the top subregions of the 2^+-*oriented* $\mathcal{H}^2_{k+n-1}$. This is because the top half of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ consists of two 2^+-*oriented* $\mathcal{H}^2_{k+n-1}$ approximations. Thus, the recurrence relation is $\psi^{\{B,F,H\}}_{2^+,n} = 2 \times \psi^{\{B,F,H\}}_{2^+,n-1}$. Since $\psi^{\{B,F,H\}}_{2^+,1} = 2$, we obtain

$$\psi^{\{B,F,H\}}_{2^+,n} = 2^n.$$

The bottom half of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ consists of a 1^+-*oriented* $\mathcal{H}^2_{k+n-1}$ and a 1^--*oriented* $\mathcal{H}^2_{k+n-1}$. In the bottom boundary subregions {C,G,I}, each 1^--*oriented* $\mathcal{H}^2_k$ in the 1^+-*oriented* $\mathcal{H}^2_{k+n-1}$ approximation becomes a 2^+-*oriented* $\mathcal{H}^2_k$ in the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ approximation; each 1^+-*oriented* $\mathcal{H}^2_k$ in the 1^--*oriented* $\mathcal{H}^2_{k+n-1}$ approximation becomes a 2^+-*oriented* $\mathcal{H}^2_k$ in the 2^+-*oriented* $\mathcal{H}^2_{k+n}$ approximation. No other than the 1^--*oriented* and 1^+-*oriented* $\mathcal{H}^2_k$ approximations in the $\mathcal{H}^2_{k+n-1}$ approximations becomes a 2^+-*oriented* $\mathcal{H}^2_k$ in the $\mathcal{H}^2_{k+n}$. Thus, it follows that

$$\psi^{\{C,G,I\}}_{2^+,n} = \psi^{\{C,G,I\}}_{1^-,n-1} + \psi^{\{C,G,I\}}_{1^+,n-1}.$$

Since there exist no 2^--*oriented* $\mathcal{H}^2_k$ approximations in the bottom boundary subregions, $\psi^{\{C,G,I\}}_{2^-,n} = 0$. Thus,

$$\psi^{\{C,G,I\}}_{2^+,n} + \psi^{\{C,G,I\}}_{1^-,n} + \psi^{\{C,G,I\}}_{1^+,n} = 2^n.$$

Similarly, on the left boundary subregion, we obtain the following recurrence relations:

$$\begin{aligned} \psi^{\{D,F,G\}}_{1^+,n} &= \psi^{\{D,F,G\}}_{2^+,n-1} + \psi^{\{D,F,G\}}_{2^-,n-1} \\ \psi^{\{D,F,G\}}_{1^+,n} + \psi^{\{D,F,G\}}_{2^+,n} + \psi^{\{D,F,G\}}_{2^-,n} &= 2^n. \end{aligned}$$

Then, from the above four recurrence relations,

$$\begin{aligned} \psi^{\{C,G,I\}}_{2^+,n} + 2\psi^{\{D,F,G\}}_{1^+,n} &= (2^{n-1} - \psi^{\{C,G,I\}}_{2^+,n-1}) + 2(2^{n-1} - \psi^{\{D,F,G\}}_{1^+,n-1}) \\ &= (2^{n-2} + \psi^{\{C,G,I\}}_{2^+,n-2}) + 2(2^{n-2} + \psi^{\{D,F,G\}}_{1^+,n-2}) \\ &= 3 \times 2^{n-2} + (\psi^{\{C,G,I\}}_{2^+,n-2} + 2\psi^{\{D,F,G\}}_{1^+,n-2}). \end{aligned}$$

Since

$$\psi^{\{C,G,I\}}_{2^+,1} + 2\psi^{\{D,F,G\}}_{1^+,1} = 2$$

and

$$\psi^{\{C,G,I\}}_{2^+,2} + 2\psi^{\{D,F,G\}}_{1^+,2} = 4,$$

we obtain

$$\psi^{\{C,G,I\}}_{2^+,n} + 2\psi^{\{D,F,G\}}_{1^+,n} = 2^n.$$

From $\psi^{\{E,H,I\}}_{1^-,n} = \psi^{\{D,F,G\}}_{1^+,n}$ due to the self-symmetry of the 2^+-*oriented* $\mathcal{H}^2_{k+n}$, it follows that

$$\psi^{\{C,G,I\}}_{2^+,n} + \psi^{\{D,F,G\}}_{1^+,n} + \psi^{\{E,H,I\}}_{1^-,n} = \psi^{\{C,G,I\}}_{2^+,n} + 2\psi^{\{D,F,G\}}_{1^+,n} = 2^n.$$

Now, consider subregions {F,G,H,I}. The $\mathcal{H}^2_k$ approximations in {F,H} are always 2^+-*oriented*, the $\mathcal{H}^2_k$ in {G} is either 2^+-*oriented* or 1^+-*oriented*, and the $\mathcal{H}^2_k$ in {I} is either 2^+-*oriented* or 1^--*oriented*. Thus, $\psi^{\{F,H\}}_{2^+,n} = 2$ and $\psi^{\{G,I\}}_{2^+,n} + \psi^{\{G,I\}}_{1^+,n} + \psi^{\{G,I\}}_{1^-,n} = 2$. Therefore,

$$\begin{aligned} \psi^{\{B\}}_{2^+,n} &= \psi^{\{B,F,H\}}_{2^+,n} - \psi^{\{F,H\}}_{2^+,n} = 2^n - 2 \\ \psi^{\{C\}}_{2^+,n} + \psi^{\{D\}}_{1^+,n} + \psi^{\{E\}}_{1^-,n} &= (\psi^{\{C,G,I\}}_{2^+,n} + \psi^{\{D,F,G\}}_{1^+,n} + \psi^{\{E,H,I\}}_{1^-,n}) \\ &\quad - (\psi^{\{G,I\}}_{2^+,n} + \psi^{\{G,I\}}_{1^+,n} + \psi^{\{G,I\}}_{1^-,n}) \\ &= 2^n - 2. \end{aligned}$$

So far, we have derived the first two equations given in this lemma.

Finally, to derive the third equation, consider subregions {B,C,D,E}. Since the total number of $\mathcal{H}^2_k$ approximations in those subregions is $4(2^n - 2)$,

$$\psi^{\{B,C,D,E\}}_{2^+,n} + \psi^{\{B,C,D,E\}}_{2^-,n} + \psi^{\{B,C,D,E\}}_{1^-,n} + \psi^{\{B,C,D,E\}}_{1^+,n} = 4(2^n - 2).$$

There exist no 2^--*oriented* $\mathcal{H}^2_k$ in {B,C}, no 1^--*oriented* $\mathcal{H}^2_k$ in {B,D}, and no 1^+-*oriented* $\mathcal{H}^2_k$ in {B,E}. That is, $\psi^{\{B,C\}}_{2^-,n} = \psi^{\{B,D\}}_{1^-,n} = \psi^{\{B,E\}}_{1^+,n} = 0$. Therefore,

$$\begin{aligned} &\psi^{\{D,E\}}_{2^+,n} + \psi^{\{D,E\}}_{2^-,n} + \psi^{\{C\}}_{1^-,n} + \psi^{\{C\}}_{1^+,n} = 4(2^n - 2) \\ &\qquad - (\psi^{\{B,C\}}_{2^+,n} + \psi^{\{B,C\}}_{2^-,n} + \psi^{\{B,D,E\}}_{1^-,n} + \psi^{\{B,D,E\}}_{1^+,n}) \\ &\qquad = 4(2^n - 2) - (\psi^{\{B,C\}}_{2^+,n} + \psi^{\{E\}}_{1^-,n} + \psi^{\{D\}}_{1^+,n}) = 2(2^n - 2). \end{aligned}$$

□

Proof of Lemma 7. Every $\mathcal{H}_k^2$ approximation in subregion {B} is 2^+-*oriented*, and there exists no 2^--*oriented* $\mathcal{H}_k^2$ approximation in subregion {C}. Thus, the number of vertical edges in subregions {B,C} is the sum of $\psi_{2^+,n}^{\{B,C\}} V_k$ and $(\psi_{1^+,n}^{\{C\}} + \psi_{1^-,n}^{\{C\}})H_k$. Likewise, the number of horizontal edges in subregions {D,E} is the sum of $(\psi_{2^+,n}^{\{D,E\}} + \psi_{2^-,n}^{\{D,E\}})H_k$ and $(\psi_{1^+,n}^{\{D\}} + \psi_{1^-,n}^{\{E\}})V_k$ because there exist no 1^--*oriented* $\mathcal{H}_k^2$ in subregion {D} and no 1^+-*oriented* $\mathcal{H}_k^2$ in subregion {E}. Thus, the total number of edges in E_2 is given by

$$\begin{aligned}&(\psi_{2^+,n}^{\{B,C\}} + \psi_{1^+,n}^{\{D\}} + \psi_{1^-,n}^{\{E\}})V_k + \\ &(\psi_{1^+,n}^{\{C\}} + \psi_{1^-,n}^{\{C\}} + \psi_{2^+,n}^{\{D,E\}} + \psi_{2^-,n}^{\{D,E\}})H_k \\ &\quad = 2(2^n - 2)(H_k + V_k) \qquad \text{(by Lemma 6)}.\end{aligned}$$

Each edge in E_2 is cut 2^k times and $H_k + V_k = 2^{2k} - 1$. Therefore,

$$N_2 = 2(2^n - 2)(2^{2k} - 1)2^k = 2(2^n - 2)2^{3k} - 2(2^n - 2)2^k.$$

□

Proof of Lemma 8. First,

$$\begin{aligned}\alpha_k + \beta_k &= \sum_{i=1}^{2^k} i h_k(i) + \sum_{i=1}^{2^k} (2^k - i + 1) h_k(i) \\ &= \sum_{i=1}^{2^k} (2^k + 1) h_k(i).\end{aligned}$$

From the definition of H_k, $H_k = \sum_{i=1}^{2^k} h_k(i)$. Therefore,

$$\alpha_k + \beta_k = (2^k + 1) H_k.$$

Second,

$$\begin{aligned}\gamma_k &= \sum_{i=1}^{2^{k-1}} i v_k(i) + \sum_{i=2^{k-1}+1}^{2^k} i v_k(i) \\ &= \sum_{i=1}^{2^{k-1}} i v_k(i) + \sum_{i=1}^{2^{k-1}} (2^{k-1} + i) v_k(2^{k-1} + i).\end{aligned}$$

Since 2-*oriented* $\mathcal{H}_k^2$ approximations are vertically self-symmetric,

$$v_k(2^k - i + 1) = v_k(i)$$

holds for any i $(1 \le i \le 2^{k-1})$. Thus,

$$\begin{aligned}\gamma_k &= \sum_{i=1}^{2^{k-1}} i v_k(i) + \sum_{i=1}^{2^{k-1}} (2^{k-1} + i) v_k(2^{k-1} - i + 1) \\ &= \sum_{i=1}^{2^{k-1}} i v_k(i) + \sum_{i=1}^{2^{k-1}} (2^k - i + 1) v_k(i).\end{aligned}$$

From the definition of V_k and self-symmetry, $V_k = 2 \sum_{i=1}^{2^{k-1}} v_k(i)$. Therefore,

$$\gamma_k = \sum_{i=1}^{2^{k-1}} (2^k + 1) v_k(i) = \frac{1}{2}(2^k + 1) V_k.$$

□

Proof of Lemma 9. In E_4, the number of horizontal cuts from a single *u-gradient*$_k$ is $2 \times \alpha_k$, the number of horizontal cuts from a single *d-gradient*$_k$ is $2 \times \beta_k$, and the number of vertical cuts from a single *s-gradient*$_k$ is $2 \times \gamma_k$. Thus,

$$\begin{aligned}N_4 &= 2\alpha_k \psi_{2^+,n}^{\{B\}} + 2\beta_k(\psi_{2^+,n}^{\{C\}} + \psi_{1^+,n}^{\{D\}} + \psi_{1^-,n}^{\{E}\}) \\ &\qquad + 2\gamma_k(\psi_{2^+,n}^{\{D,E\}} + \psi_{2^-,n}^{\{D,E\}} + \psi_{1^-,n}^{\{C\}} + \psi_{1^+,n}^{\{C\}}) \\ &= 2\alpha_k(2^n - 2) + 2\beta_k(2^n - 2) + 4\gamma_k(2^n - 2) \quad \text{(by Lemma 6)} \\ &= 2(2^n - 2)(\alpha_k + \beta_k + 2\gamma_k) \\ &= 2(2^n - 2)(2^k + 1)(H_k + V_k) \qquad \text{(by Lemma 8)} \\ &= 2(2^n - 2)(2^k + 1)(2^{2k} - 1).\end{aligned}$$

In E_5, the number of horizontal cuts from a single *u-gradient*$_k$ is α_k, the number of horizontal cuts from a single *d-gradient*$_k$ is β_k, and the number of vertical cuts from a single *s-gradient*$_k$ is γ_k. Thus,

$$N_5 = 2\alpha_k + 2\beta_k + 4\gamma_k = 2(2^k + 1)(2^{2k} - 1).$$

□

ACKNOWLEDGMENTS

This work was sponsored in part by US National Science Foundation CAREER Award IIS-9876037 and grants IRI-9625428 and DMS-9873442, by the Advanced Research Projects Agency under contract No. DABT63-94-C-0049. It was also supported by US National Science Foundation Cooperative Agreement No. IRI-9411299 and by DARPA/ITO through Order F463, issued by ESC/ENS under contract N66001-97-C-851. Additional funding was provided by donations from NEC and Intel. The authors assume all responsibility for the contents of the paper.

REFERENCES

[1] D.J. Abel and D.M. Mark, "A Comparative Analysis of Some Two-Dimensional Orderings," *Int'l J. Geographical Information Systems,* vol. 4, no. 1, pp. 21–31, Jan. 1990.

[2] J.J. Bartholdi and L.K. Platzman, "An O($n \log n$) Travelling Salesman Heuristic Based on Spacefilling Curves," *Operation Research Letters,* vol. 1, no. 4, pp. 121–125, Sept. 1982.

[3] M. Berger, *Geometry II.* New York: Springer-Verlag, 1987.

[4] T. Bially, "Space-Filling Curves: Their Generation and Their Application to Bandwidth Reduction," *IEEE Trans. Information Theory,* vol. 15, no. 6, pp. 658–664, Nov. 1969.

[5] A.R. Butz, "Convergence with Hilbert's Space Filling Curve," *J. Computer and System Sciences,* vol. 3, pp. 128–146, 1969.

[6] I.S. Duff, "Design Features of a Frontal Code for Solving Sparse Unsymmetric Linear Systems Out-of-Core," *SIAM J. Scientific Statistical Computing,* vol. 5, no. 2, pp. 270–280, June 1984.

[7] C.R. Dyer, "The Space Efficiency of Quadtrees," *Computer Graphics and Image Processing,* vol. 19, no. 4, pp. 335–348, Aug. 1982.

[8] C. Faloutsos, "Multiattribute Hashing Using Gray Codes," *Proc. 1986 ACM SIGMOD Conf.,* pp. 227–238, May 1986.

[9] C. Faloutsos, "Analytical Results on the Quadtree Decomposition of Arbitrary Rectangles," *Pattern Recognition Letters,* vol. 13, no. 1, pp. 31–40, Jan. 1992.

[10] C. Faloutsos, H.V. Jagadish, and Y. Manolopoulos, "Analysis of the N-Dimensional Quadtree Decomposition for Arbitrary Hyperrectangles," *IEEE Trans. Knowledge and Data Eng.,* vol. 9, no. 3, pp. 373–383, May/June 1997.

[11] C. Faloutsos and S. Roseman, "Fractals for Secondary Key Retrieval," *Proc. ACM Principles of Database Systems Conf.,* pp. 247–252, Mar. 1989.

[12] P.J. Giblin, *Graphs, Surfaces, and Homology,* second ed. New York: Chapman and Hall, 1981.

[13] D. Hilbert, "Über die stetige Abbildung einer Linie auf Flächenstück," *Math. Ann.,* vol. 38, pp. 459–460, 1891.

[14] H.V. Jagadish, "Linear Clustering of Objects with Multiple Attributes," *Proc. ACM SIGMOD Conf.,* pp. 332–342, May 1990.

[15] H.V. Jagadish, "Spatial Search with Polyhedra," *Proc. Sixth Int'l Conf. Data Eng.,* pp. 311–319, Feb. 1990.

[16] H.V. Jagadish, "Analysis of the Hilbert Curve for Representing Two-Dimensional Space," *Information Processing Letters,* vol. 62, no. 1, pp. 17–22, Apr. 1997.

[17] M. Kaddoura, C.-W. Ou, and S. Ranka, "Partitioning Unstructured Computational Graphs for Nonuniform and Adaptive Environments," *IEEE Parallel and Distributed Technology,* vol. 3, no. 3, pp. 63–69, Fall 1995.

[18] A. Lempel and J. Ziv, "Compression of Two-Dimensional Images," *NATO ASI Series,* vol. F12, pp. 141–154, June 1984.

[19] J. Orenstein, "Spatial Query Processing in an Object-Oriented Database System," *Proc. 1986 ACM SIGMOD Conf.,* pp. 326–336, May 1986.

[20] E.A. Patrick, D.R. Anderson, and F.K. Bechtel, "Mapping Multi-Dimensional Space to One Dimension for Computer Output Display," *IEEE Trans. Computers,* vol. 17, no. 10, pp. 949–953, Oct. 1968.

[21] G. Peano, "Sur une Courbe qui Remplit Toute une Aire Plane," *Math. Ann.,* vol. 36, pp. 157–160, 1890.

[22] R.L. Rivest, "Partial Match Retrieval Algorithms," *SIAM J. Computing,* vol. 5, no. 1, pp. 19–50, Mar. 1976.

[23] Y. Rong and C. Faloutsos, "Analysis of the Clustering Property of Peano Curves," Technical Report CS-TR-2792, UMIACS-TR-91-151, Univ. of Maryland, Dec. 1991.

[24] J.B. Rothnie and T. Lozano, "Attribute Based File Organization in a Paged Memory Environment," *Comm. ACM,* vol. 17, no. 2, pp. 63–69, Feb. 1974.

[25] C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *IEEE Computer,* vol. 27, no. 3, pp. 17–28, Mar. 1994.

[26] H. Sagan, "A Three-Dimensional Hilbert Curve," *Int'l J. Math. Ed. Science Technology,* vol. 24, pp. 541–545, 1993.

[27] C.A. Shaffer, "A Formula for Computing the Number of Quadtree Node Fragments Created by a Shift," *Pattern Recognition Letters,* vol. 7, no. 1, pp. 45–49, Jan. 1988.

[28] G.F. Simmons, *Introduction to Topology and Modern Analysis.* New York: McGraw-Hill Book Company, Inc., 1963.

Bongki Moon received the the BS and MS degrees in computer engineering from Seoul National University, Korea, in 1983 and 1985, respectively, and the PhD degree in computer science from the University of Maryland, College Park, in 1996. He is an assistant professor in the Department of Computer Science, University of Arizona. His current research interests include high-performance spatial databases, scalable web servers, data mining and warehousing, and parallel and distributed processing. He worked for Samsung Electronics Corp., Korea, as a member of the research staff at the Communication Systems Division from 1985 to 1990.

H.V. Jagadish received the PhD degree from Stanford University in 1985. He spent over a decade at AT&T Bell Laboratories in Murray Hill, N.J., eventually becoming head of AT&T Labs database research department at the Shannon Laboratory in Florham Park, New Jersey. He has also served as a professor at the University of Illinois in Urbana-Champaign. Currently, he is a professor of computer science at the University of Michigan in Ann Arbor. Dr. Jagadish is well-known for his broad-ranging research on databases and is currently the founding editor of the *ACM SIGMOD Digital Review*. Among the many professional positions he has held, he has been an associate editor for the *ACM Transactions on Database Systems* (1992-1995) and program chair of the ACM SIGMOD annual conference (1996). The focus of his current work is the design of hierarchical databases for highly distributed and heteregenous contexts.

Christos Faloutsos received the BSc degree in electrical engineering (1981) from the National Technical University of Athens, Greece, and the MSc and PhD degrees in computer science from the University of Toronto, Canada. Dr. Faloutsos is currently a faculty member at Carnegie Mellon University (CMU) in Pittsburgh, Pennsylvania. Prior to joining CMU, he was on the faculty of the Department of Computer Science at University of Maryland in College Park. He has spent sabbaticals at IBM Almaden and AT&T Bell Labs. He has worked as a consultant to several companies, including AT&T Research, Lucent, and SUN. He received the Presidential Young Investigator Award by the US National Science Foundation (1989), two "best paper" awards (SIGMOD '94, VLDB '97), and four teaching awards. He has published more than 70 refereed articles, one monograph, and has filed for four patents. His research interests include physical database design, searching methods for text, geographic information systems indexing methods for multimedia databases, and data mining. He is a member of the IEEE and the IEEE Computer Society.

Joel H. Saltz received the BS degree in mathematics and physics from University of Michigan and the MA degree in mathematics from University of Michigan in Ann Arbor, in 1977 and 1978, respectively, and the MD and PhD degrees in computer science from Duke University in 1985. He is a professor with the Department of Computer Science and the Institute for Advanced Computer Studies, University of Maryland, College Park. He leads a research group whose goal is to develop methods to produce portable compilers that generate efficient multiprocessor code for irregular scientific problems that are unstructured, sparse, adaptive, or block structured. He is a member of the IEEE and the IEEE Computer Society.