# Analysis of the Clustering Properties of Hilbert Space-filling Curve*

*Bongki Moon*[‡] *H.V. Jagadish*[§] *Christos Faloutsos*[‡†] *Joel H. Saltz*[‡]

[‡] *Institute for Advanced Computer Studies and*
*Department of Computer Science*
*University of Maryland*
*College Park, MD 20742*
$\{$*bkmoon,christos,saltz*$\}$*@cs.umd.edu*

[§] *AT&T Research*
*Murray Hill, NJ 07974*
*jag@research.att.com*

## Abstract

*Several schemes for linear mapping of multidimensional space have been proposed for many applications such as access methods for spatio-temporal databases, image compression and so on. In all these applications, one of the most desired properties from such linear mappings is clustering, which means the locality between objects in the multidimensional space is preserved in the linear space. It is widely believed that the Hilbert space-filling curve achieves the best clustering [1, 13]. In this paper we provide closed-form formulas of the number of clusters required by a given query region of an arbitrary shape (e.g., polygons and polyhedra) for Hilbert space-filling curve. Both the asymptotic solution for a general case and the exact solution for a special case generalize the previous work [13], and they agree with the empirical results that the number of clusters depends on the hyper-surface area of the query region and not on its hyper-volume. We have also shown that Hilbert curve achieves better clustering than the z-curve [21]. From the practical point of view, the formulas given in this paper provide a simple measure which can be used to predict the required disk access behaviors and hence the total access time.*

**Index Terms**: locality-preserving linear mapping, range queries, multi-attribute 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 such a total ordering is found for a given spatial or multi-attribute database, one can use any one-dimensional access method such as a 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 [17]. The idea is to develop a single numeric index on a one-dimensional space for each point in 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 [22] to buckets stored on disk. The linear traversal specifies the order in which the objects

[*] This work was supported in part by the Advanced Research Projects Agency under contract No. DABT63-94-C-0049, the National Science Foundation under contract No. NSF ASC9318183, CDR-8803012, EEC-94-02384, IRI-9205273 and IRI-8958546 (PYI), with matching funds from EMPRESS Software Inc. and Thinking Machines Corp. The authors assume all responsibility for the contents of the paper.

[†] Dr. Christos Faloutsos is also affiliated with Institute for Systems Research, University of Maryland, College Park, MD 20742.

***Submitted to IEEE Transactions on Knowledge and Data Engineering, March 1996.***
***If accepted, copyright will transfer to IEEE.***
***CS-TR-3611 and UMIACS-TR-96-20. Available at URL http://www.cs.umd.edu/TR/UMCP-CSD:CS-TR-3611***

are fetched from disk as well as the number of blocks fetched. The number of non-consecutive disk accesses will be determined by the order of blocks fetched. Although in the range query the order of blocks fetched is not explicitly specified, it is reasonable to assume that the set of blocks fetched can be rearranged into a number of groups of consecutive blocks by database server or disk controller mechanism [23]. Since it is preferred to fetch a set of consecutive disk blocks rather than a randomly scattered set 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 space. A good clustering of multidimensional points on the one-dimensional sequence of disk blocks will 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 the mapping which preserves locality:

1. In traditional databases, a multi-attribute space must be mapped into a one-dimensional space to allow efficient handling of partial-match queries [20]; in numerical analysis, large multidimensional arrays [5] have to be stored on disk, which is a linear structure.

2. In image compression, a family of methods use the mapping to transform the image into a bit string; subsequently, any standard compression method can be applied [16]. A good clustering of pixels will result in fewer long runs of similar pixel values, thus 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 the mapping. For example, for the travelling salesman problem, the cities are linearly ordered and visited accordingly [2].

5. Locality-preserving mappings are used for bandwidth reduction of digitally sampled signals [3] and for graphics display generation [18].

6. In scientific parallel processing, locality-preserving linearization techniques are preferred for dynamic unstructured mesh partitioning [15].

Sophisticated mapping functions have been proposed in the literature. One, based on interleaving bits from the coordinates, which is called z-ordering was proposed in [17]. Its improvement was suggested by Faloutsos in [7], using Gray coding on the interleaved bits. A third method, based on the Hilbert curve [11], has been proposed in [9]. In the mathematical context, these three mapping functions are based on different space-filling curves: *z-curve*, *Gray code* with bit-interleaving and *Hilbert curve*, respectively. Figure 1 illustrates linear orderings yielded by the space-filling curves for $4 \times 4$ grids.

In [13] we have studied the mapping functions from multidimensional space to one-dimensional space, and showed that under most circumstances the mapping based on Hilbert space-filling curve outperforms the others. 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 hyper-rectangle or polyhedron in multidimensional space.

For purposes of analysis, we assume 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 and so on. Here we shall use instead the average number of *clusters* or *continuous runs* of grid points within a subspace represented by a given query, as the measure of clustering performance of the Hilbert space-filling curve. If each grid point is mapped to one disk block, this measure exactly corresponds to the number of non-consecutive disk accesses, which involve additional seek time. It is also highly correlated

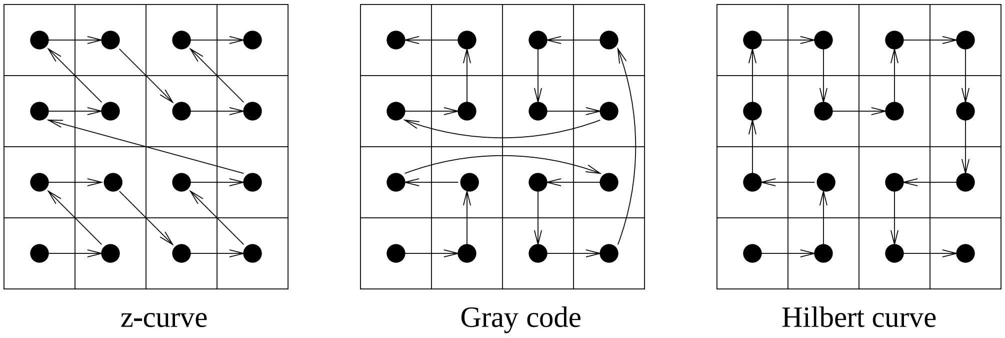


Figure 1: Illustration of space-filling curves

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 [13].

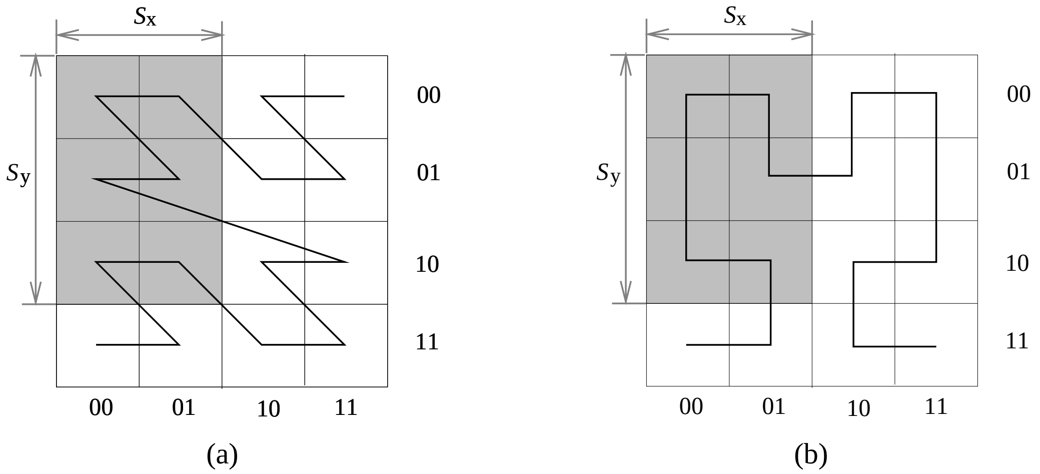


Figure 2: Illustration of clusters: (a) two clusters for z-curve, (b) one cluster for Hilbert curve

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

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

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

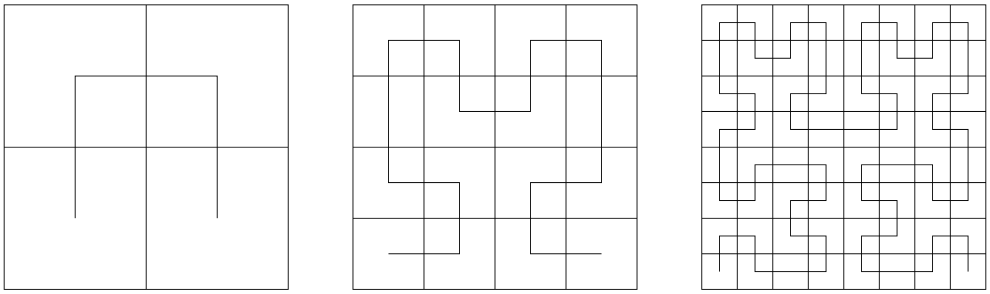
The definition of the *d*-dimensional rectilinear polyhedron is given in Section 3. Note that in the *d*-dimensional space with finite granularity, for any *d*-dimensional object such as spheres, ellipsoids, quadric cones and so on, there exists a corresponding (rectilinear) polyhedron that contains 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 *2*-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

G. Peano, in 1890, discovered the existence of a continuous curve which passes through every point of a closed square [19]. 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* [26]. 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 G. Peano discovered the first space-filling curve, it was D. Hilbert in 1891 who was the first to recognize a general geometric procedure that allows the construction of an entire class of space-filling curve [11]. 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 $n = 1, 2, 3, \ldots$ 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. Figure 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 [24].



(a) First step (b) Second step (c) Third step

Figure 3: The first three steps of Hilbert space-filling curve

Note that Hilbert gave the space-filling curve, in a geometric form only, for mapping $I$ into $S$ (*i.e.,* 2-dimensional Euclidean space). Generation of a *3*-dimensional Hilbert curve was described in [13, 24]. A generalization of Hilbert curves, in an analytic form, for higher dimensional space was given in [4]. In this paper, *d*-dimensional Euclidean space with finite granularity is of our interest. Thus, we use *the $k$-th order approximation* of *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 $k$-th order approximation of 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 Hilbert curve in Figure 3 correspond to $\mathcal{H}_1^2$, $\mathcal{H}_2^2$ and $\mathcal{H}_3^2$, respectively.

In [13], we have compared clustering properties of several space mapping functions by considering only $2\times2$ range queries. Among z-curve (2.625), Gray coding (2.5) and Hilbert curve (2), Hilbert curve was the best in minimizing the number of clusters. The numbers within the parentheses are the average number of clusters for $2\times2$ range queries. Rong and Faloutsos [21] derived a closed form expression of the average number of clusters for the z-curve, which gives 2.625 for $2\times2$ range queries (exactly the same with the result given in [13]) 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 [12] derived closed form expressions of the average number of clusters for the Hilbert curve in a *2*-dimensional grid using $2\times2$ and $3\times3$ square regions only. 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 a conclusion that Hilbert ordering deserves closer attention as an alternative to z-curve ordering.

Closely related analysis for the average number of *d*-dimensional quadtree nodes has been presented in the literature. Dyer in [6] presented an analysis for the best, worst and average case of a square of size $2^n\times2^n$, giving an approximate formula for the average case. Shaffer in [25] 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 the formula for the average number of blocks for such squares (averaged over all the possible positions). In [8, 10], we generalized some of these formulae for arbitrary 2-dimensional and *d*-dimensional rectangles.

## 3 Asymptotic Analysis

In this section, we give an asymptotic formula of the clustering property of Hilbert space-filling curves for general polyhedra in *d*-dimensional space. The symbols used in this section are summarized in Table 1. The polyhedra we consider here are not necessarily convex but *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 perpendicular to one of the $d$ coordinate axes, which is a subset of $\mathbf{R}^d$ and homeomorphic to (d-1)-dimensional sphere $S^{d-1}$.*

For $d = 2$ the set $V$ is, by definition, a *Jordan curve,* 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 Hilbert curve into (or from) the polyhedron.*

Thus, we expect that the number of clusters is approximately proportional to the perimeter or surface area of the *d*-dimensional polyhedron ($d \geq 2$). Coupled 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 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 *d*-dimensional Euclidean space, that is, approximately one $d$-th of the edges are aligned to the $i$-th 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

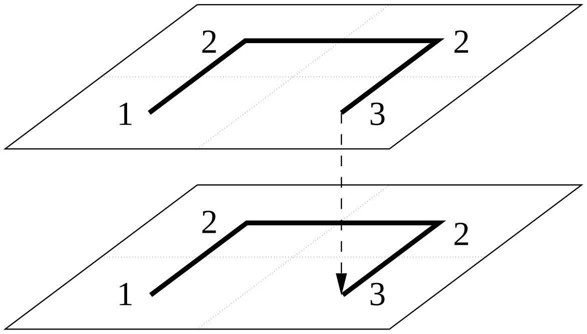


Figure 4: 3-dimensional Hilbert curve (template of $\mathcal{H}_k^3$)

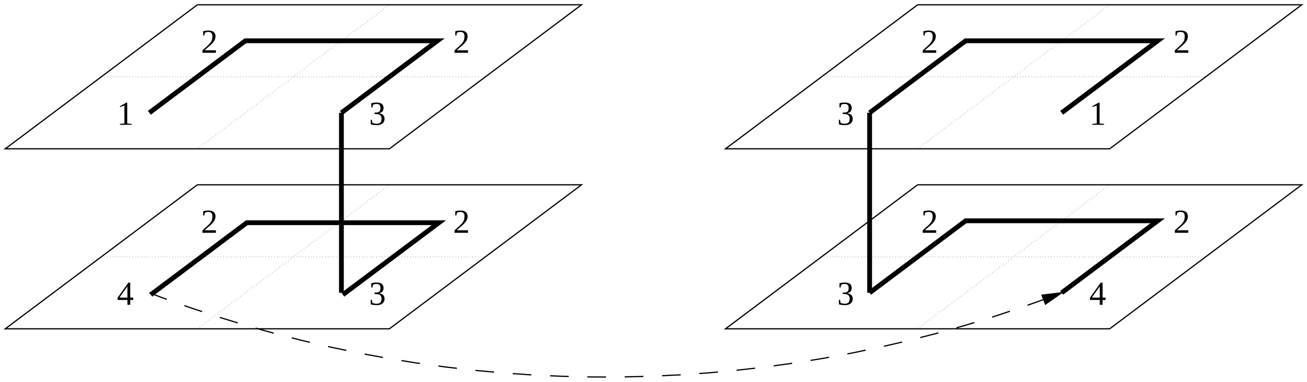


Figure 5: 4-dimensional Hilbert curve (template of $\mathcal{H}_k^4$)

distribution of the edges provides key leverage for deriving the asymptotic solution. To show the uniform distribution, it is important to understand

- how the $k$-th order approximation of Hilbert curve is derived from lower order approximations, and
- how the *d*-dimensional Hilbert curve is extended from the *2*-dimensional Hilbert curve, which was described only in geometric form in [11]. Analytic forms for *d*-dimensional Hilbert curve were presented in [4].

In a *d*-dimensional space, $\mathcal{H}_k^d$ is derived from $\mathcal{H}_1^d$ by replacing each vertex in $\mathcal{H}_1^d$ 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 the number of vertices of $\mathcal{H}_1^d$ is $2^d$, $\mathcal{H}_k^d$ is composed of $2^d$ $\mathcal{H}_{k-1}^d$'s and $(2^d - 1)$ edges each connecting two of them.

Before describing the extension for *d*-dimensional Hilbert curve, we define *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 Hilbert curve starts its traversal, the coordinates of the start and end vertices of $\mathcal{H}_1^d$ differ only in one dimension, which means both the vertices lie on a line parallel to one of $d$ coordinate axes. From now on we say a $\mathcal{H}_1^d$ is *i-oriented* if its start and end vertices lie on a line parallel to the $i$-th 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 $\mathcal{H}_k^d$ is derived.

In the following we examine the process that generates $\mathcal{H}_k^d$ from $\mathcal{H}_k^{d-1}$. Figure 4 and Figure 5 illustrate the generation of $\mathcal{H}_k^3$ from $\mathcal{H}_k^2$, and $\mathcal{H}_k^4$ from $\mathcal{H}_k^3$, respectively. Each vertex of the curves represents rotated and/or reflected $\mathcal{H}_{k-1}^3$ in Figure 4 and $\mathcal{H}_{k-1}^4$ in Figure 5, and is annotated by a number indicating its orientation. In general, when the $d$-th dimension is added to a *(d-1)*-dimensional Hilbert curve, each vertex of $\mathcal{H}_1^{d-1}$ (that is, $\mathcal{H}_{k-1}^{d-1}$) 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 $d$-th dimensional axis. For example, in Figure 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 $(2^{d-1}-1)$ $\mathcal{H}_{k-1}^d$'s remain unchanged, they are all *j-oriented* for some $j$ $(1 \leq j < d)$.

Then the whole $2^{d-1}$ $\mathcal{H}_{k-1}^d$'s are replicated by reflection and finally the two replicas are connected by an edge parallel to the $d$-th coordinate axis (called *d-oriented* edge) to form a *d-oriented* $\mathcal{H}_k^d$. In short, whenever a dimension (say, the $d$-th dimension) is added, two *d-oriented* $\mathcal{H}_{k-1}^d$'s are introduced, the number of *1-oriented* $\mathcal{H}_{k-1}^d$'s remains unchanged as two, and the number of $\mathcal{H}_{k-1}^d$'s of the other orientations are doubled.

| 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 *d*-dimensional Hilbert curve |
| $\varphi_i$ | Number of *i-oriented* $\mathcal{H}_{k-1}^d$'s 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 rectilinearly polyhedral query $q$ |
| $\mathcal{N}_d$ | Average number of clusters within a given *d*-dimensional rectilinear polyhedron |

Table 1: Definition of Symbols

The following lemma provides a ground for leading to a more interesting Lemma 2, which is useful in deriving the asymptotic formula.

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

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

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

*Proof.* It can be proven by induction on $d$. □

In the following lemma, we show that the edges of $d$ different orientations approaches 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 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 $\varepsilon_{i,k}$'s and $\varphi_i$'s. As we mentioned previously, the fundamental operations involved in expanding 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 $\mathcal{H}_{k-1}^d$ 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, any two distinct $\mathcal{H}_{k-1}^d$'s 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 *1-oriented* edges in $\mathcal{H}_k^d$ consists of $2^{d-1}$ connection edges in $\mathcal{H}_1^d$, *d-oriented* edges in *1-oriented* $\mathcal{H}_{k-1}^d$'s, *(d-1)-oriented* edges in *2-oriented* $\mathcal{H}_{k-1}^d$'s, *(d-2)-oriented* edges in *3-oriented* $\mathcal{H}_{k-1}^d$'s 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 $\varphi_i$ are in Lemma 1. 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 $y$ only in one dimension by unity. In other words, the coordinates of the neighbors that lie in a line parallel to the $i$-th 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 $i^+$*-neighbor* and $i^-$*-neighbor* of $y$, respectively.

Butz showed in [4] that any unit increment in Hilbert order produces a unit increment in one of $d$ coordinates and leaves the other $d-1$ coordinates unchanged. The implication is that, for any grid point $y$, both the neighbors of $y$ in the linear ordering imposed by Hilbert curve are chosen from $2d$ neighbors of $y$ in the *d*-dimensional grid space. Of the two neighbors of $y$ in Hilbert ordering, the one closer to the start end of Hilbert traversal is called *predecessor* of $y$.

*Notation 3.3 For a grid point $y$ in 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 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 $d$-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 Hilbert mapping, it follows that for any $i$ $(1 \leq i \leq d)$ the probability that $\overline{yz}$ is parallel to the $i$-th dimensional axis is $d^{-1}$. This implies that the probability that $z$ is either $i^+$*-neighbor* or $i^-$*-neighbor* of $y$ is $d^{-1}$. The proof is now complete. □

The *d*-dimensional rectilinear polyhedra of our interest can be of arbitrary shape; the number and size of 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 $i^+$*-surface*.
- If a point $y$ is inside the polyhedron and its $i^-$*-neighbor* is outside, then the point $y$ faces $i^-$*-surface*.

For example, Figure 6 illustrates grid points which face surfaces in *2*-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, the grid points A and D face $1^+$-*surface*, and the grid point B (on the convex) faces both $1^+$-*surface* and $2^+$-*surface*. Although the 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 the grid point B is approximately twice that through the grid point A (or D). There is no chance that the Hilbert curve enters through the grid point C.

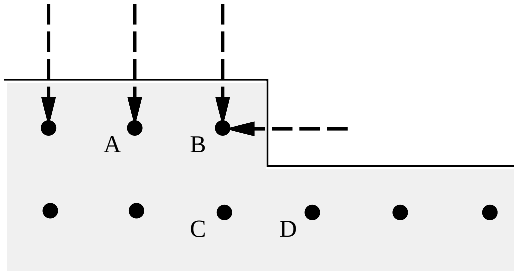


Figure 6: Illustration of grid points facing surfaces

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 without proof an elementary remark.

*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 $\mathcal{S}_q$ be the total surface area of a given rectilinearly polyhedral query $q$. Then,*

$$\lim_{k\rightarrow\infty} \mathcal{N}_d = \frac{\mathcal{S}_q}{2d} \tag{3}$$

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

$$\begin{aligned} \lim_{k\rightarrow\infty} \mathcal{N}_d &= \sum_{i=1}^{d} (S_i^+ p_i^+ + S_i^- p_i^-) \\ &= \sum_{i=1}^{d} S_i^+ (p_i^+ + p_i^-) \qquad \text{(by Remark 3.2)} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{d} S_i^{+} \frac{1}{d} && \text{(by Lemma 3)} \\
&= \frac{\mathcal{S}_q}{2d}.
\end{aligned}
$$

The proof is complete. □

Theorem 1 confirms our early conjecture that the number of clusters is approximately proportional to the surface area of a *d*-dimensional polyhedron, and provides $(2d)^{-1}$ as the constant factor of a linear function. In *2*-dimensional space, the average number of clusters for z-curve approaches one third of the perimeter of the query rectangle plus two thirds of the side length of the rectangle in the unfavored direction [21]. Now it comes clear that Hilbert curve achieves better clustering than z-curve because the average number of clusters for 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:*

*(i) Given an $s_1 \times s_2 \times \cdots \times s_d$ hyper-rectangle,* $\lim_{k\to\infty} \mathcal{N}_d = \frac{1}{d}\sum_{i=1}^{d}(\frac{1}{s_i}\prod_{j=1}^{d} s_j)$.

*(ii) 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(ii) provides 2 as an average number of clusters, which is exactly the same with the result given in [13].

# 4 Exact Analysis : A special case

In this section, we give a closed-form exact formula for the average number of clusters in *2*-dimensional space. Specifically, we assume that grid space is mapped by $\mathcal{H}_{k+n}^2$ and query regions are square of size $2^k \times 2^k$. We first describe our approach and then the formal derivation of the solution is presented in the following lemmas and a theorem. Table 2 summarizes the symbols used in this section.

## 4.1 Basic concepts

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

*Remark 4.1 The number of clusters within a given 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 square windows of all possible positions. Then the average number of clusters is simply obtained by dividing this number by twice the number of possible positions of the 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.*

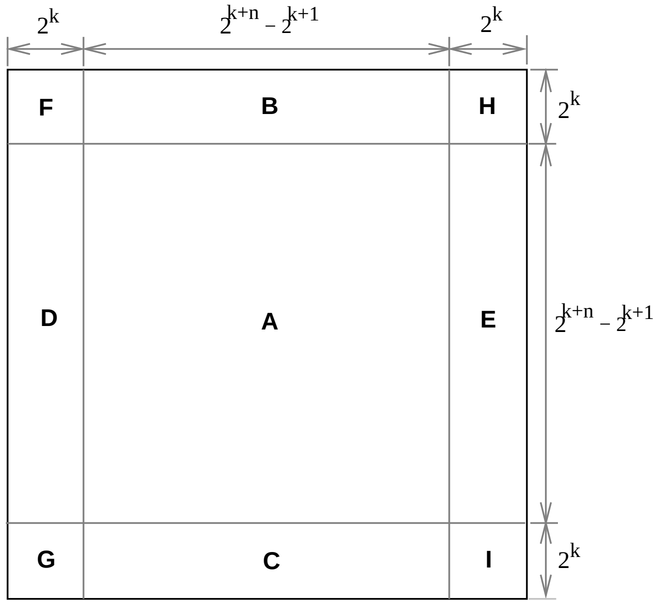


Figure 7: $\mathcal{H}^2_{k+n}$ divided into nine subregions

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 less number of square windows can cut the edges near the boundary. Thus it is useful to consider a $2^{k+n} \times 2^{k+n}$ grid region $\mathcal{H}^2_{k+n}$ as a collection of $2^{2n}$ $\mathcal{H}^2_k$'s each of which is connected to one or two neighbors by connection edges. From now on, we mean by an *internal edge* one of $2^{2k} - 1$ edges in a $\mathcal{H}^2_k$, and by a *connection edge* one connecting two $\mathcal{H}^2_k$'s.

We divide the grid region $\mathcal{H}^2_{k+n}$ into nine subregions as depicted in Figure 7. The width of the subregions on the boundary is $2^k$. Then, for example, subregion F includes only one $\mathcal{H}^2_k$ and connected to subregions B and D by a horizontal connection edge and a vertical connection edge, respectively. Subregion B includes $(2^n - 2)$ $\mathcal{H}^2_k$'s connected to each other by $(2^n - 3)$ horizontal connection edges inclusive to the subregion, and connected to subregions F and H by two other horizontal connection edges straddling the boundaries of subregions.

Now consider an edge (either an internal or a connection edge) near the center of subregion A, and a horizontal edge in subregion B. The edge in the subregion A can be cut by $2^{k+1}$ square windows whose positions within the region are mutually distinct. On the other hand, the horizontal edge in the subregion B can be cut by different number of distinct windows depending on the position of the edge. Specifically, if the edge 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:

A1. Every edge (either horizontal or vertical) at least one of whose end points reside in subregion A is cut $2^{k+1}$ times.

A2. Every vertical edge in subregions B and C is cut $2^k$ times by top or bottom sides of windows.

A3. Every horizontal edge in subregions D and E is cut $2^k$ times by left or right sides of windows.

A4. Every connection edge in subregions $\{$B, F, H$\}$ is horizontal and resides in the $2^k$-th row from the topmost and hence cut $2^{k+1}$ times by left and right sides of windows. Every connection edge in subregions $\{$C, G, I$\}$ is horizontal and resides in the $2^k$-th row from the topmost and hence cut twice by left and right sides of windows.

A5. Every connection edge in subregions $\{$D, F, G$\}$ is vertical and resides in the first column from the leftmost and hence cut twice by top and bottom sides of windows. Every connection edge in subregions $\{$E, H, I$\}$ is vertical and resides in the first column from the rightmost and hence cut twice by top and bottom sides of windows.

A6. Every horizontal edge in the $i$-th row from the topmost of the subregion B is cut $2 \times i$ times by both left and right sides of windows, and every horizontal edge in the $i$-th row from the topmost of the subregion C is cut $2^{k+1} - 2 \times i + 2$ times by both left and right sides of windows.

A7. Every vertical edge in the $i$-th column from the leftmost of the subregion D is cut $2 \times i$ times by both top and bottom sides of windows, and every vertical edge in the $i$-th column from the leftmost of the subregion E is cut $2^{k+1} - 2 \times i + 2$ times by both top and bottom sides of windows.

A8. Every horizontal edge in the $i$-th row from the topmost of subregions $\{\mathtt{F},\mathtt{H}\}$ is cut $i$ times by either left or right sides of windows.

A9. Every horizontal edge in the $i$-th row from the topmost of subregions $\{\mathtt{G},\mathtt{I}\}$ is cut $2^k - i + 1$ times by either left or right sides of windows.

A10. Every vertical edge in the $i$-th column from the leftmost of subregions $\{\mathtt{F},\mathtt{G}\}$ is cut $i$ times by either top or bottom sides of windows.

A11. Every vertical edge in the $i$-th column from the leftmost of subregions $\{\mathtt{H},\mathtt{I}\}$ is cut $2^k - i + 1$ times by either top or bottom sides of windows.

A12. 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 within the $\mathcal{H}^2_{k+n}$ grid region into the following five groups:

(i) $E_1$: a group of edges described in the observations A1. Each edge is cut $2^{k+1}$ times.

(ii) $E_2$: a group of edges described in the observations A2 and A3. Each edge is cut $2^k$ times.

(iii) $E_3$: a group of edges described in the observations A4 and A5. Each connection edge on the top boundary (*i.e.*, subregions $\{\mathtt{B},\mathtt{F},\mathtt{H}\}$) is cut $2^{k+1}$ times and any other connection edge is cut twice.

(iv) $E_4$: a group of edges described in the observations A6 to A7. 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).

(v) $E_5$: a group of edges described in the observations A8 to A11. 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* $E_i$*.*

Within the $\mathcal{H}^2_{k+n}$ region, the number of all possible positions of $2^k \times 2^k$ windows is $(2^{k+n} - 2^k + 1)^2$. Since, in addition to $N_1, \ldots, N_5$, there are two more cuts from observation A12, 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 following, we give closed-form expressions for individual edge groups $N_1, \ldots, N_5$.

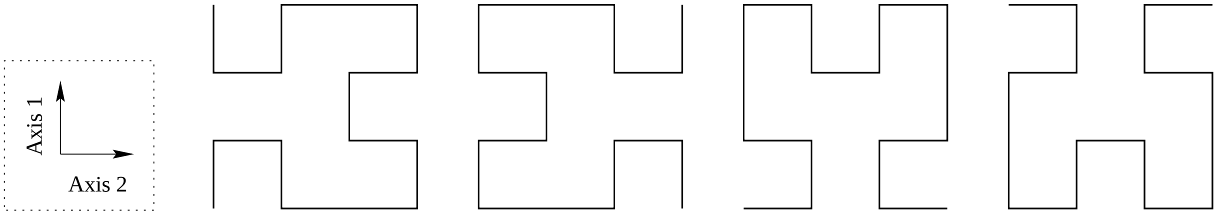
## 4.2 Formal derivation

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

*Notation 4.3 An i-oriented* $\mathcal{H}^d_k$ *is called i*$^+$*-oriented (or i*$^-$*-oriented) if the* $i$*-th coordinate of its end point is greater (or less) than that of its start point.*

| 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* $\mathcal{H}^2_k$'s in the subregion $R$ of a *$2^+$-oriented* $\mathcal{H}^2_{k+n}$ |
| $\psi^{\{R\}}_{i^-,n}$ | Number of *$i^-$-oriented* $\mathcal{H}^2_k$'s 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 |

Table 2: Definition of Symbols



(a) *$1^+$-oriented* (b) *$1^-$-oriented* (c) *$2^+$-oriented* (d) *$2^-$-oriented*

Figure 8: Four different orientations of $\mathcal{H}^2_2$

Figure 8 illustrates *$1^+$-oriented*, *$1^-$-oriented*, *$2^+$-oriented* and *$2^-$-oriented* $\mathcal{H}^2_2$'s, respectively from left to right. Note here that the vertical axis is considered as the first dimensional axis and the horizontal axis is considered as the second dimensional axis.

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 straddling the boundaries between subregion A and the other subregions, which is not quite straightforward, as well as the number of edges inclusive to the subregion A. We approach this with counting the number of edges in the complementary set $\overline{E_1}$ (that is, $\{$edges in $\mathcal{H}^2_{k+n}\} - E_1$). Since $\overline{E_1}$ consists of edges in $4(2^n - 1)$ $\mathcal{H}^2_k$'s in boundary subregions B through I and connection edges in $E_3$, $|\overline{E_1}|$ is equal to $4(2^n - 1) \times (2^{2k} - 1) + |E_3|$. 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 given grid region is *$2^+$-oriented* $\mathcal{H}^2_{k+n}$.

*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 $\mathcal{H}^2_{k+n}$, respectively.*

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

**Lemma 4** *For any positive integer $n$,*

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

*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 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*

$$\begin{aligned} N_1 &= 2(2^n - 2)^2 2^{3k} + 3(2^n - 2)2^k && (6)\\ N_3 &= 2^{n+k} + 4(2^n - 1) && (7) \end{aligned}$$

*Proof.* $\mathcal{H}^2_{k+n}$ and $\mathcal{H}^2_k$ contain $2^{2(k+n)} - 1$ and $2^{2k} - 1$ edges, respectively. Since the number of $\mathcal{H}^2_k$'s in the boundary subregions is $4(2^n - 1)$, the total number of edges in $E_1$ is given by

$$(2^{2(k+n)} - 1) - 4(2^n - 1)(2^{2k} - 1) - (5 \times 2^{n-1} - 2) = 2^{2k}(2^n - 2)^2 + 3(2^{n-1} - 1).$$

From the fact that each edge in $E_1$ is cut $2^{k+1}$ times, it follows that

$$N_1 = 2^{k+1}(2^{2k}(2^n - 2)^2 + 3(2^{n-1} - 1)) = 2(2^n - 2)^2 2^{3k} + 3(2^n - 2)2^k.$$

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

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

□

Now 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 $\mathcal{H}^2_k$ is determined by its orientation, it is necessary to find the number of $\mathcal{H}^2_k$'s of different orientations in the subregions {`B,C,D,E`}. In the following, we give notations for the number of horizontal and vertical edges in a $\mathcal{H}^2_k$, and the number of $\mathcal{H}^2_k$'s of different orientations in the boundary subregions in Figure 7.

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

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

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

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

$$\begin{aligned} \psi^{\{B\}}_{2^+,n} &= 2^n - 2 && (8)\\ \psi^{\{D\}}_{1^+,n} + \psi^{\{E\}}_{1^-,n} + \psi^{\{C\}}_{2^+,n} &= 2^n - 2 && (9)\\ \psi^{\{C\}}_{1^+,n} + \psi^{\{C\}}_{1^-,n} + \psi^{\{D,E\}}_{2^+,n} + \psi^{\{D,E\}}_{2^-,n} &= 2(2^n - 2). && (10) \end{aligned}$$

*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*

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

*Proof.* Every $\mathcal{H}_k^2$ in subregion $\texttt{B}$ is *2*$^+$*-oriented*, and no *2*$^-$*-oriented* $\mathcal{H}_k^2$ exists in subregion $\texttt{C}$. Thus the number of vertical edges in subregions $\{\texttt{B,C}\}$ 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 $\{\texttt{D,E}\}$ 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 no *1*$^-$*-oriented* $\mathcal{H}_k^2$ exists in the subregions $\texttt{D}$ and no *1*$^+$*-oriented* $\mathcal{H}_k^2$ exists in the subregions $\texttt{E}$. Thus, the total number of edge in $E_2$ is given by

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

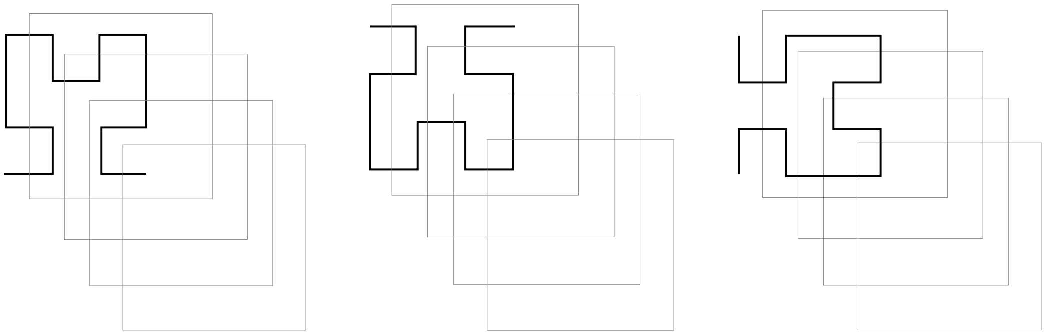
Each edge in $E_2$ is cut $2^k$ times and $H_k + V_k = 2^{2k} - 1$. Therefore,

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

□

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

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



(a) *u-gradient*$_2$ (b) *d-gradient*$_2$ (c) *s-gradient*$_2$

Figure 9: Three different gradients and cutting windows

To derive closed-form expressions of $N_4$ and $N_5$, we first give the definitions for different types of gradients. Consider *2*$^+$*-oriented* $\mathcal{H}_k^2$'s in subregions $\{\texttt{B,C,D,E}\}$. From the observations A6 and A7, the number of cuts from horizontal edges in a *2*$^+$*-oriented* $\mathcal{H}_k^2$ in the subregion $\texttt{B}$ is $\sum_{i=1}^{2^k} 2ih_k(i)$. Likewise, the number of cuts from horizontal edges in a *$2^+$-oriented* $\mathcal{H}_k^2$ in the subregion $\mathtt{C}$ is $\sum_{i=1}^{2^k} 2(2^k - i + 1)h_k(i)$, and the number of cuts from vertical edges in a *$2^+$-oriented* $\mathcal{H}_k^2$ in the subregion $\mathtt{D}$ or $\mathtt{E}$ is $\sum_{i=1}^{2^k} 2iv_k(i)$. The reason the number of cuts from vertical edges is the same in both the subregions $\mathtt{D}$ and $\mathtt{E}$ is 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* (i) *A $2^+$-oriented $\mathcal{H}_k^2$ is called u-gradient$_k$ if its horizontal edges in the $i$-th row from the topmost are cut $i$ or $2i$ times.*

(ii) *A $2^+$-oriented $\mathcal{H}_k^2$ is called d-gradient$_k$ if its horizontal edges in the $i$-th row from the topmost are cut $2^k - i + 1$ or $2(2^k - i + 1)$ times.*

(iii) *A $2^+$-oriented $\mathcal{H}_k^2$ is called s-gradient$_k$ if its vertical edges in the $i$-th column from the leftmost (or rightmost) are cut $i$ or $2i$ times.*

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

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

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

*Proof.* Given in Appendix A. □

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

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

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

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

Thus, in the subregions $\{\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E}\}$, the number of *u-gradient$_k$*'s is $\psi_{2+,n}^{\{B\}}$, the number of *d-gradient$_k$*'s is $\psi_{2+,n}^{\{C\}} + \psi_{1+,n}^{\{D\}} + \psi_{1-,n}^{\{E\}}$, and the number of *s-gradient$_k$*'s is $\psi_{2+,n}^{\{D,E\}} + \psi_{2-,n}^{\{D,E\}} + \psi_{1-,n}^{\{C\}} + \psi_{1+,n}^{\{C\}}$. In the subregions $\{\mathtt{F},\mathtt{G},\mathtt{H},\mathtt{I}\}$, the number of *u-gradient$_k$*'s is two, the number of *d-gradient$_k$*'s is two, and the number of *s-gradient$_k$*'s is four. From this observation and Lemma 6 and Lemma 8, it follows that

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

$$\begin{aligned} N_4 &= 2(2^n - 2)(2^k + 1)(2^{2k} - 1) && (13) \\ N_5 &= 2(2^k + 1)(2^{2k} - 1) && (14) \end{aligned}$$

*Proof.* 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,

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

In $E_5$, the number of horizontal cuts from a single *u-gradient*$_k$ is $\alpha_k$, the number of horizontal cuts from a single *d-gradient*$_k$ is $\beta_k$, and the number of vertical cuts from a single *s-gradient*$_k$ is $\gamma_k$. Thus, $N_5 = 2\alpha_k + 2\beta_k + 4\gamma_k = 2(2^k+1)(2^{2k}-1)$. □

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*

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

*Proof.* From Equation (4),

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

□

In the limit as $n$ grows large, $\mathcal{N}_2(k, k+n)$ 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(ii) for $d = 2$.

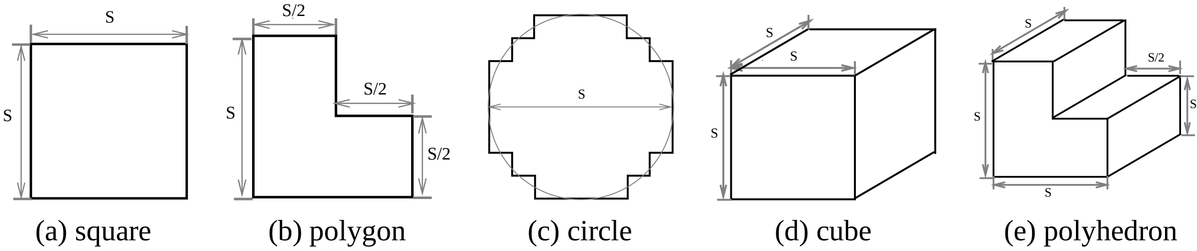
# 5 Experimental Results

To demonstrate the correctness of the asymptotic and exact analyses presented in the previous sections, we carried out simulation experiments for query regions of various sizes and shapes in both 2-dimensional and 3-dimensional grid spaces.

### Arrangements of experiments

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 provides excellent approximation for general $d$-dimensional query regions of arbitrary sizes and shapes as well as showing the correctness of the exact solution for 2-dimensional $2^k \times 2^k$ query regions. To obtain exact measurements of actual number of clusters, we averaged the number of clusters within query regions of 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 $2^k \times 2^k$ query squares.

However, the number of all possible queries is exponential on the dimensionality. Consequently, for a large grid space and high dimensionality, each simulation run may require processing an excessively large number of queries, making the simulation take too long. Thus, in our experiments, we limited the dimensionality of the grid space to two and three. For query shapes, we chose squares, concave polygons and circles for 2-dimensional



(a) square (b) polygon (c) circle (d) cube (e) polyhedron

Figure 10: Illustration of sample query shapes

cases, and cubes, concave polyhedra and spheres for *3*-dimensional cases. Figure 10 illustrates some of those query shapes used in our experiments.

Theorem 1 only states that as the size of grid space grows the average number of clusters approaches half the surface area of a given query region divided by the dimensionality. It does not provide details as to how rapidly the number of clusters converges to the asymptotic solution. To address this, we repeated the same set of simulation runs over grids of different sizes $N \times N$ (or $N \times N \times N$) with $N = 32, 40, 48, 56, 64, 128$. The side length $s$ of square or cubic queries and the bounding boxes of the other query shapes was varied from 1 to 32 for both 2-dimensional and 3-dimensional cases.

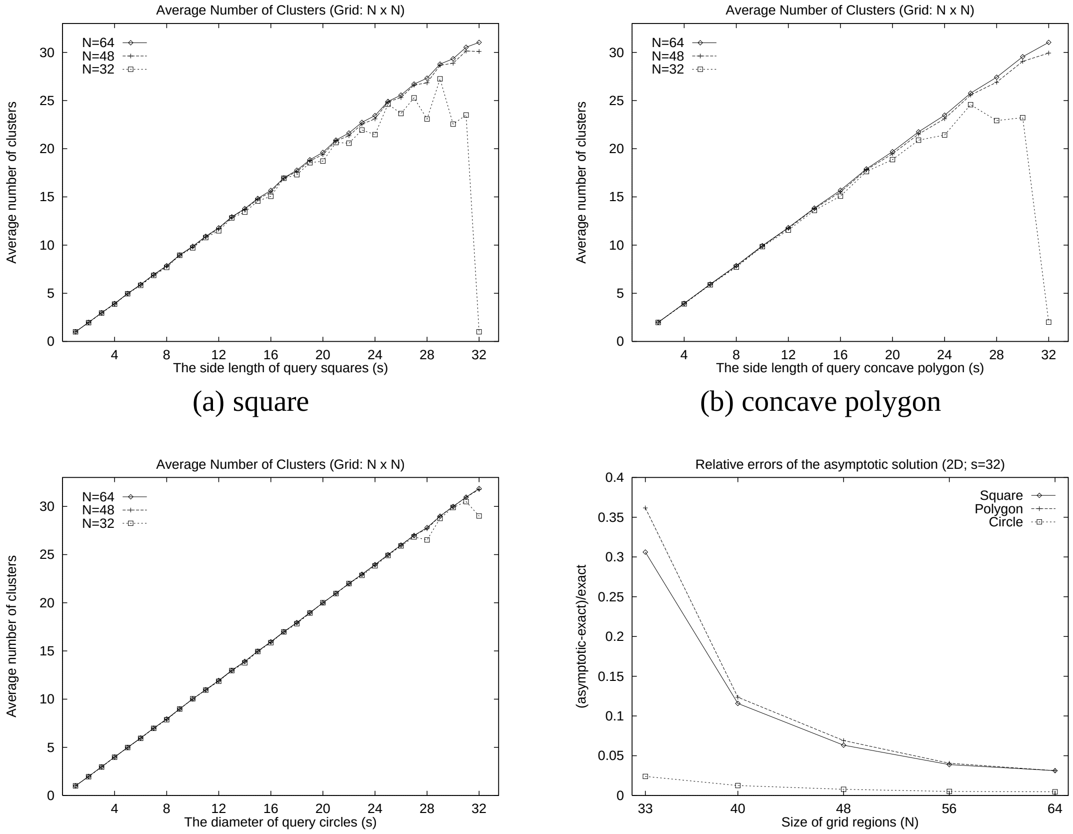
### Results

The first set of experiments were carried out in a 2-dimensional grid space. Figure 11(a)-(c) shows the measured average number of clusters for query regions of squares, concave polygons and circles, respectively. The sizes and shapes of the query regions are illustrated in Figure 10(a)-(c). To minimize confusion, only the results for grid size $N = 32/48/64$ have been shown. Figure 11(d) gives the relative errors of the asymptotic solution given in Theorem 1 for a fixed query size $s = 32$. Note that, in Figure 11(d), we used $N = 33$ instead of $N = 32$ to avoid the cases where the query region and grid are identical and hence the asymptotic solution is far away from its corresponding exact number. Such situations are shown in Figure 11(a) and (b). When $s = N = 32$, it is obvious that the number of clusters is exactly one for the square query region, and exactly two for the concave polygonal query region, while the asymptotic solution is 32.

With a few exceptional cases where $s$ is very close to $N$, the number of clusters forms a linear curve for each query shape and is almost identical for the three 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. This should not be surprising because they have the same perimeter for a given $s$. For example, we can always find a rectilinear polygon that contains the same set of grid points as a given circle of diameter $s$, and it is always the case that the perimeter of the rectilinear polygon is equal to that of a square of side length $s$. (See Figure 10(c).) In general, in a *2*-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 fact justifies the general approach of using the minimum bounding rectangle of a given query region because it does not increase the actual number of clusters (*i.e.*, the number of non-consecutive disk accesses).

It is interesting to see that the average number of clusters for circular query regions is very close to the asymptotic solution even when $s$ approaches to $N$, and the relative error is always far smaller than those of the other query shapes. It is also observed that the measured numbers of clusters shown in Figure 11(a) for square query regions of side length power of two exactly match the exact solution in Theorem 2 when $N$ is also a power of two.

The same set of experiments were carried out in a 3-dimensional grid space. Figure 12(a)-(c) shows the measured average number of clusters for query regions of cubes, concave polyhedra and spheres, and Figure 12(d)



(a) square

(b) concave polygon

(c) circle

(d) relative error

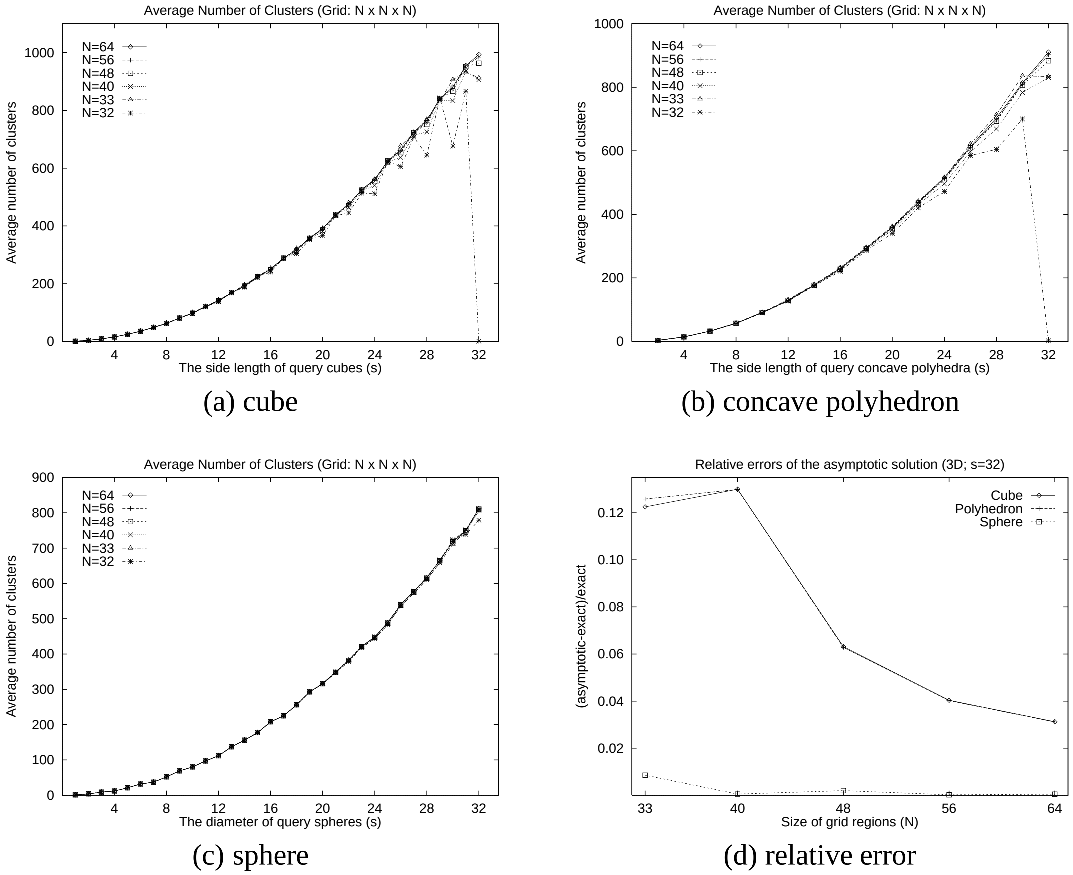
Figure 11: Average number of clusters and relative error of asymptotic solution

gives the relative errors of the asymptotic solution for a fixed query size $s = 32$. Note again that, in Figure 12(d), we used $N = 33$ instead of $N = 32$ for the same reason as in the previous 2-dimensional simulation experiments. Like the 2-dimensional case, similar trends are observed in both the average number of clusters and relative errors for all the three query shapes. The number of clusters forms a quadratic curve for each query shape, and the relative error for spheres is far smaller than that for the other query shapes. However, if we look closer, the constant factors of the quadratic functions are slightly different among different query shapes.

To determine the quadratic functions for each query shape, we applied the least-square curve fitting method to the results from grid of size $N = 64$. The approximate quadratic functions were obtained as follows:

$$\begin{aligned} f_a(s) &= 0.973818s^2 + 0.354112s - 1.309880 \\ f_b(s) &= 0.883308s^2 + 0.471050s - 1.975170 \\ f_c(s) &= 0.78435s^2 + 0.112668s + 0.768710. \end{aligned}$$

The approximate function $f_a(s)$ for cubic query regions confirms the asymptotic solution given in Corollary 1(ii) because it is quite close to $s^2$. In contrast, $f_b(s)$ and $f_c(s)$ the functions for concave polyhedral and spherical query regions are much lower than that. The reason is that, unlike the 2-dimensional case, the surface area of a concave polyhedron or a sphere is smaller than that of its minimum bounding hyper-rectangle. For example, the surface area of the polyhedron illustrated in Figure 10(e) is $\frac{11}{2}s^2$ while that of the corresponding cube is $6s^2$. The surface area of the rectilinear polyhedron that contains the same set of grid points inside a sphere of diameter $s = 32$ is 4872, which is far smaller than $6 \times 32^2$ grid points for the corresponding cube ($s = 32$). Note that the



(a) cube

(b) concave polyhedron

(c) sphere

(d) relative error

Figure 12: Average number of clusters and relative error of asymptotic solution

coefficients of the quadratic terms in $f_b(s)$ and $f_c(s)$ are fairly close to $\frac{11}{12}$ and $\frac{4872}{6\times32^2}$, respectively.

This indicates that, in a *d*-dimensional space ($d \geq 3$), accessing the minimum bounding hyper-rectangle of a given query region may incur additional non-consecutive disk accesses, and hence supports the argument made in [14] that the minimum bounding rectangle may not be a good approximation to a non-rectangular object.

The main conclusions from our experiments are:

- The exact solution given in Theorem 2 matches exactly the experimental results for square queries of size $2^k \times 2^k$.
- The asymptotic solutions given in Theorem 1 and Corollary 1 provide excellent approximation for $d$-dimensional queries of arbitrary shapes and sizes. For example, when $d = 3, N = 64$, and $s = 32$, the relative errors were less than 4 percent for cubic and polyhedral queries and less then 1 percent for spherical queries.
- Assuming that blocks are arranged on disk by Hilbert ordering, accessing the minimum bounding rectangles of *d*-dimensional ($d \geq 3$) query regions may increase the number of non-consecutive accesses, whereas this is not the case for *2*-dimensional queries.

# 6 Conclusions

We have studied the clustering property of the Hilbert space-filling curve as a linear mapping of multidimensional space. Through algebraic analysis we have provided simple formulas which 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:

- Our result presented in Theorem 2 generalizes the previous work done only for $2 \times 2$ query regions [13] by providing an exact closed-form formula for $2^k \times 2^k$ for any $k$ $(k \geq 1)$.
- The asymptotic solution given in Theorem 1 further generalizes it for *d*-dimensional polyhedral query regions.
- We have shown that, in a *2*-dimensional space, the Hilbert curve achieves better clustering than the z-curve; the 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 [21]. We conjecture that this trend will hold even in higher dimensional spaces.
- We have shown that accessing the minimum bounding hyper-rectangles of *d*-dimensional ($d \geq 3$) non-rectangular query regions may incur extra overhead by adding to the number of clusters (*i.e.*, non-consecutive disk accesses).

From the practical point of view, it is important to predict and minimize the number of clusters because it determines the number of non-consecutive disk accesses, which in turn incur additional seek time. Assuming that blocks are arranged on disk by Hilbert ordering, now we can provide a simple measure comprising only the perimeter or surface area of a given query region and its dimensionality, which can then be used to predict the required disk access behaviors and hence the total access time. Future work includes the extension of the exact analysis for *d*-dimensional space.

# A Appendix: proofs

**Lemma 4** *For any positive integer* $n$,

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

*Proof.* A *2*$^+$*-oriented* $\mathcal{H}^2_{k+n}$ is composed of four $\mathcal{H}^2_{k+n-1}$'s and three connection edges. Two $\mathcal{H}^2_{k+n-1}$'s on the top half are *2*$^+$*-oriented* and two $\mathcal{H}^2_{k+n-1}$'s on the bottom half are *1*$^+$*-oriented* on the left and *1*$^-$*-oriented* on the right, respectively. Among the three edges connecting the four $\mathcal{H}^2_{k+n-1}$'s, 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 hence included in the boundary subregion of the $\mathcal{H}^2_{k+n}$. Thus the main observations are:

(i) The number of connection edges in top boundary subregion of $\mathcal{H}^2_{k+n}$ is the sum of those in top boundary subregions of two *2*$^+$*-oriented* $\mathcal{H}^2_{k+n-1}$'s.

(ii) The number of connection edges in bottom boundary subregion of $\mathcal{H}^2_{k+n}$ is the sum of those in bottom boundary subregions of a *1*$^+$*-oriented* $\mathcal{H}^2_{k+n-1}$ and a *1*$^-$*-oriented* $\mathcal{H}^2_{k+n-1}$.

(iii) The number of connection edges in left (or right) boundary subregion of $\mathcal{H}^2_{k+n}$ is the sum of those in left (or right) boundary subregions of a *2*$^+$*-oriented* $\mathcal{H}^2_{k+n-1}$ and a *1*$^+$*-oriented* (or *1*$^-$*-oriented*) $\mathcal{H}^2_{k+n-1}$ 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* $\mathcal{H}^2_{k+n-1}$ and so on, 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 $b_n + 2s_n = 2(2^n - 1)$. □

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

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

*Proof.* Consider a *2⁺-oriented* $\mathcal{H}^2_{k+n}$, which is composed of four $\mathcal{H}^2_{k+n-1}$'s and three connection edges. The number of *2⁺-oriented* $\mathcal{H}^2_{k}$'s in the subregions {B,F,H} of the *2⁺-oriented* $\mathcal{H}^2_{k+n}$ is twice the number of *2⁺-oriented* $\mathcal{H}^2_{k}$'s in the subregions {B,F,H} of the *2⁺-oriented* $\mathcal{H}^2_{k+n-1}$ because the top half of the *2⁺-oriented* $\mathcal{H}^2_{k+n}$ contains two *2⁺-oriented* $\mathcal{H}^2_{k+n-1}$'s. 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* $\mathcal{H}^2_{k+n}$ contains a *1⁺-oriented* $\mathcal{H}^2_{k+n-1}$ and a *1⁻-oriented* $\mathcal{H}^2_{k+n-1}$. Thus, on the bottom boundary subregions {C,G,I}, each *1⁻-oriented* $\mathcal{H}^2_{k}$ in the $\mathcal{H}^2_{k+n-1}$'s turns a *1⁻-oriented* $\mathcal{H}^2_{k}$ and a *2⁺-oriented* $\mathcal{H}^2_{k}$ in the *2⁺-oriented* $\mathcal{H}^2_{k+n}$; each *1⁺-oriented* $\mathcal{H}^2_{k}$ in the $\mathcal{H}^2_{k+n-1}$'s turns a *2⁺-oriented* $\mathcal{H}^2_{k}$ and a *1⁺-oriented* $\mathcal{H}^2_{k}$ in the *2⁺-oriented* $\mathcal{H}^2_{k+n}$. No subgrid other than *1⁻-oriented* $\mathcal{H}^2_{k}$'s and *1⁺-oriented* $\mathcal{H}^2_{k}$'s in the $\mathcal{H}^2_{k+n-1}$'s turns *2⁺-oriented* $\mathcal{H}^2_{k}$'s in the $\mathcal{H}^2_{k+n}$. Thus it follows that

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

In addition, $\psi^{\{C,G,I\}}_{2^-,n} = 0$ because no *2⁻-oriented* $\mathcal{H}^2_{k}$ exist on the bottom boundary subregions. 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 $\psi_{2^+,1}^{\{C,G,I\}} + 2\psi_{1^+,1}^{\{D,F,G\}} = 2$ and $\psi_{2^+,2}^{\{C,G,I\}} + 2\psi_{1^+,2}^{\{D,F,G\}} = 4$, we obtain

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

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

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

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

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

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

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

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

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

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

□

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

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

*Proof.* First, $\alpha_k + \beta_k = \sum_{i=1}^{2^k} ih_k(i) + \sum_{i=1}^{2^k}(2^k - i + 1)h_k(i) = \sum_{i=1}^{2^k}(2^k + 1)h_k(i)$. 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, $\gamma_k = \sum_{i=1}^{2^{k-1}} iv_k(i) + \sum_{i=2^{k-1}+1}^{2^k} iv_k(i) = \sum_{i=1}^{2^{k-1}} iv_k(i) + \sum_{i=1}^{2^{k-1}}(2^{k-1} + i)v_k(2^{k-1} + i)$. Since *2-oriented* $\mathcal{H}_k^2$'s are vertically self-symmetric, $v_k(2^k - i + 1) = v_k(i)$ holds for any $i$ $(1 \leq i \leq 2^{k-1})$. Thus, $\gamma_k = \sum_{i=1}^{2^{k-1}} iv_k(i) + \sum_{i=1}^{2^{k-1}}(2^{k-1} + i)v_k(2^{k-1} - i + 1) = \sum_{i=1}^{2^{k-1}} iv_k(i) + \sum_{i=1}^{2^{k-1}}(2^k - i + 1)v_k(i)$. From the definition of $V_k$ and self-symmetry, $V_k = 2\sum_{i=1}^{2^{k-1}} v_k(i)$. Therefore,

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

□

# References

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

[2] J.J. Bartholdi and L.K. Platzman. An O($n$ log $n$) travelling salesman heuristic based on spacefilling curves. *Operation Research Letters*, 1(4):121–125, September 1982.

[3] T. Bially. Space-filling curves: Their generation and their application to bandwidth reduction. *IEEE Trans. on Information Theory*, IT-15(6):658–664, November 1969.

[4] Arthur R. Butz. Convergence with Hilbert's space filling curve. *Journal of Computer and System Sciences*, 3:128–146, 1969.

[5] I.S. Duff. Design features of a frontal code for solving sparse unsymmetric linear systems out-of-core. *SIAM J. Sci. Stat. Computing*, 5(2):270–280, June 1984.

[6] C.R. Dyer. The space efficiency of quadtrees. *Computer Graphics and Image Processing*, 19(4):335–348, August 1982.

[7] C. Faloutsos. Multiattribute hashing using gray codes. *Proc. ACM SIGMOD*, pages 227–238, May 1986.

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

[9] C. Faloutsos and S. Roseman. Fractals for secondary key retrieval. *Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 247–252, March 1989.

[10] Christos Faloutsos, H.V. Jagadish, and Yannis Manolopoulos. Analysis of the n-dimensional quadtree decomposition for arbitrary hyper-rectangles. CS-TR-3381, UMIACS-TR-94-130, Univ. of Maryland, December 1994. submitted for publication.

[11] D. Hilbert. $\ddot{U}$ber die stetige Abbildung einer Linie auf Fl$\ddot{a}$chenst$\ddot{u}$ck. *Math. Annln.*, 38:459–460, 1891.

[12] H. V. Jagadish. Analysis of the Hilbert curve for representing two-dimensional space. Technical report, AT&T Research, Murray Hill, NJ 07974, August 1996. Under review at Information Processing Letter.

[13] H.V. Jagadish. Linear clustering of objects with multiple attributes. *ACM SIGMOD Conf.*, pages 332–342, May 1990.

[14] H.V. Jagadish. Spatial search with polyhedra. *Proc. Sixth IEEE Int'l Conf. on Data Engineering*, February 1990.

[15] Maher Kaddoura, Chao-Wei Ou, and Sanjay Ranka. Partitioning unstructured computational graphs for nonuniform and adaptive environments. *IEEE Parallel and Distributed Technology*, 3(3):63–69, Fall 1995.

[16] A. Lempel and J. Ziv. Compression of two-dimensional images. *NATO ASI Series*, F12:141–154, June 1984.

[17] J. Orenstein. Spatial query processing in an object-oriented database system. *Proc. ACM SIGMOD*, pages 326–336, May 1986.

[18] Edward A. Patrick, Douglas R. Anderson, and F. K. Bechtel. Mapping multidimensional space to one dimension for computer output display. *IEEE Transactions on Computers*, C-17(10):949–953, October 1968.

[19] G. Peano. Sur une courbe qui remplit toute une aire plaine. *Math. Ann.*, 36, 1890.

[20] R.L. Rivest. Partial match retrieval algorithms. *SIAM J. Comput*, 5(1):19–50, March 1976.

[21] Yi Rong and Christos Faloutsos. Analysis of the clustering property of Peano curves. Techn. Report CS-TR-2792, UMIACS-TR-91-151, Univ. of Maryland, December 1991.

[22] J.B. Rothnie and T. Lozano. Attribute based file organization in a paged memory environment. *CACM*, 17(2):63–69, February 1974.

[23] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, March 1994.

[24] Hans Sagan. A three-dimensional Hilbert-Curve. *Inter. J. Math. Ed. Sc. Tech.*, 24:541–545, 1993.

[25] C.A. Shaffer. A formula for computing the number of quadtree node fragments created by a shift. *Pattern Recognition Letters*, 7(1):45–49, January 1988.

[26] George F. Simmons. *Introduction to Topology and Modern Analysis*. McGraw-Hill Book Company, Inc., New Work, 1963.