TECHNICAL RESEARCH REPORT

Analysis of the n-dimensional quadtree decomposition for arbitrary hyper-rectangles

by C. Faloutsos, H.V. Jagadish,
Y. Manolopoulos

T.R. 94-86

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Analysis of the n-dimensional quadtree decomposition for arbitrary hyper-rectangles

Christos Faloutsos *
Dept. of Computer Science and
Inst. for Systems Research (ISR)
University of Maryland,
College Park, MD 20742

H. V. Jagadish
AT&T Bell Laboratories
Murray Hill, NJ 07974

Yannis Manolopoulos †
Department of Informatics,
Aristotle University,
Thessaloniki, Greece 54006

Abstract

We give a closed-form expression for the average number of n-dimensional quadtree nodes ('pieces' or 'blocks') required by an n-dimensional hyper-rectangle aligned with the axes. Our formula includes as special cases the formulae of previous efforts for 2-dimensional spaces [8]. It also agrees with theoretical and empirical results that the number of blocks depends on the hyper-surface of the hyper-rectangle and not on its hyper-volume. The practical use of the derived formula is that it allows the estimation of the space requirements of the n-dimensional quadtree decomposition. Quadtrees are used extensively in 2-dimensional spaces (geographic information systems and spatial databases in general), as well in higher dimensionality spaces (as oct-trees for 3-dimensional spaces, e.g. in graphics, robotics and 3-dimensional medical images [2]). Our formula permits the estimation of the space requirements for data hyper-rectangles when stored in an index structure like a (n-dimensional) quadtree, as well as the estimation of the search time for query hyper-rectangles. A theoretical contribution of the paper is the observation that the number of blocks is a piece-wise linear function of the sides of the hyper-rectangle.

Categories and Subject Descriptors: E.2 [**Data Structures**]: Trees; H.2.2 [**Database Management**]: Physical Design - *access methods*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling - *Hierarchy and geometric transformations*;
General Terms: Computational Geometry, Performance, Theory.
Additional Keywords: regular decomposition, geometric data, quadtrees, oct-trees, GIS, robotics.

*His work was partially supported by the National Science Foundation (IRI-8958546, IRI-9205273), with matching funds from EMPRESS Software Inc. and Thinking Machines Inc.

†Research performed with the Dept. of Computer Science and the Inst. for Systems Research (ISR) at the University of Maryland, College Park, MD

1 Introduction

Hierarchical decomposition of space plays an important role in every application that involves geometric data. The idea is that the space is decomposed recursively into smaller and smaller pieces, until the content of each such piece is homogeneous. The problem solved in this paper is the analytical estimation of the number of pieces that an n-dimensional rectangle (hyper-rectangular region) is decomposed into.

Consider a **2-dimensional image** represented as a $2^k \times 2^k$ array of 1×1 squares. Each such square is called a **pixel**. The length $K = 2^k$ of the side of the image is called the **granularity** of the image. A geometric object within such an image is represented by turning the appropriate pixels to black, while the background is considered white. More than one geometric object may exist in an image. A **block** is a $2^m \times 2^m$ square ($0 \leq m \leq k$) obtained as the result of recursive decomposition of the image into quadrants and sub-quadrants. We focus on representing one object only. An object within an image is decomposed into blocks as in Figure 1. For example, in this figure the square [0,2]×[2,4] is a block, while the square [1,3]×[2,4] is not.

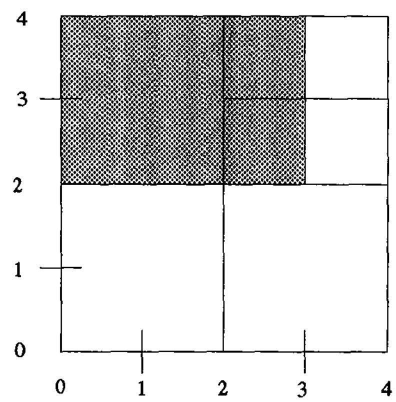

Figure 1: The shaded rectangle is decomposed in three blocks.

Such a hierarchical decomposition approach has been used in several areas, including:

- In graphics and robotics (3-dimensional space) [3, 20].
- In geographic information systems and spatial databases. The TIGER project at the U.S. Bureau of Census uses a linear quadtree representation to store all the points of interest in the map of U.S.A. [23]. A similar approach has also been used by Shaffer in the QUILT system for geographic and spatial databases [22], as well as by Orenstein in the extensible data base management system PROBE [18].
- In traditional databases, where records with n attributes correspond to points in an n-dimensional space. Many methods have been suggested to store such a collection of data, utilizing the hierarchical decomposition approach (e.g. k-d trees [4], quadtrees and their variations [11]).

- In spatio-temporal and scientific databases, where time introduces one more axis [16].
- In image databases, e.g., [2], where 3-dimensional brain scans have to be stored. Regions in these brain scans can be encoded using oct-trees, to save space and to achieve faster response on range queries.
- In Grand-Challenge databases [5] (e.g., with meteorological, environmental, sensor data e.t.c.). In general, these databases contain large multi-dimensional arrays, (e.g., tuples of the form $(x, y, z, t,$ *temperature*)) which can be stored in some multi-resolution, hierarchical fashion, clustering related (i.e., nearby) points together.
- Whenever a transformation is used (e.g., a 2-dimensional rectangle corresponds to a 4-dimensional point [9, 12]; a polyhedron is mapped to a high-dimensionality point [14]).

The problem we examine here is the following:

Given a hyper-rectangle of size $s_1 \times s_2 \times ... s_n$,

Find the number of blocks that it will span on the average.

Previous attempts have been restricted to 2-dimensional rectangles: Dyer in [6] 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 in [21] gives a closed formula for the exact number of blocks that such a square requires when anchored at a given position (x, y); he also gives the formula for the average number of blocks for such squares (averaged over all the possible positions). In a previous paper [8], we generalized some of these formulae for arbitrary (2-dimensional) rectangles. Analysis of the closely-related Peano and Hilbert space filling curves for 2-dimensional spaces was presented in [15] and [19].

In this paper, we generalize the formulae for n-dimensional rectangles. The derived formulae are useful whenever a hierarchical decomposition is used for higher-dimensionality spaces, either for data hyper-rectangles, or for query hyper-rectangles. In all these cases, the number of pieces that a hyper-rectangle decomposes into clearly affects the space overhead and the search time. Therefore, it is essential for query optimization in spatial/temporal databases [1].

The proposed methodology is as follows:

1. Find the formulae when the sides of the hyper-rectangles are of the form $2^{m_i} - 1$, for every dimension $i = 1, 2, ..., n$. Let's call these hyper-rectangles **magic**. One important observation is the fact that the solution for magic rectangles is simple.
2. Prove that the formula for a non-magic hyper-rectangle can be derived by a linear interpolation from the surrounding magic hyper-rectangles.

The paper is organized as follows. Section 2 gives some preliminary definitions and examples. Section 3 gives the solution (closed-form formulae) for the magic hyper-rectangles. Section 4 establishes a theorem that the solution for non-magic hyper-rectangles can be derived by using linear interpolation. Section 5 gives closed formulae for the expected number of blocks in the case of 2-dimensional rectangles and 3-dimensional parallelepipeds. Section 6 makes some observations and suggests future research directions.

2 Preliminaries

Symbol	Definition
n	Number of dimensions
$x_1, ..., x_n$	Co-ordinates of the lowest corner of the
	hyper-rectangle (i.e., the one closest to the origin)
s_i	Length of the hyper-rectangle in i-th dimension
$b(x_1, s_1, ..., x_n, s_n)$	Number of blocks to cover a specific hyper-rectangle
$\bar{b}(s_1, s_2, ..., s_n)$	Average number of blocks to cover the hyper-rectangle of the query size
$K = 2^k$	Granularity = side of the 'universe' in hyper-pixels

Table 1: Definition of Symbols

A hyper-rectangle is represented as $(x_1, s_1, x_2, s_2, ..., x_n, s_n)$ where x_i $(i = 1, ..., n)$ is the i-th coordinate of the **anchor** (i.e., the corner with the smallest coordinate values or the 'lower left' corner; this is the corner closest to the origin, since all the coordinates are non-negative) and s_i is the size of the hyper-rectangle on the i-th dimension. Table 1 shows the symbols and their definitions.

Definition 1. The average number of blocks for a rectangle of sides $(s_1, s_2, ..., s_n)$ is given by:

$$\bar{b}(s_1, s_2, ..., s_n) = \frac{1}{K^n} \sum_{x_1=0}^{K-1} \cdots \sum_{x_n=0}^{K-1} b(x_1, s_1, x_2, s_2, ..., x_n, s_n) \tag{1}$$

where $K = 2^k$ is the granularity. Intuitively, we let the hyper-rectangle go to each and every possible position, and we average the number of blocks that the hyper-rectangle decomposes into. Notice that:

- K should be large enough so that the $K \times K ... \times K$ hyper-cube completely encloses the hyper-rectangle under examination. In other words: $s_i \leq K$ for $i = 1, ..., n$.
- The hyper-rectangle wraps around the edges. This assumption has been used in all the previous analyses of quadtrees [6, 8].

Some important observations, that allow recursive decomposition of the problem:

Observation 1 - 'Slicing'. If a hyper-rectangle starts at an odd number, then we can 'slice off' the left hyper-plane. In such a case, the number of blocks of the two pieces added together is the same as the number of blocks of the whole hyper-rectangle, in this given position. Without loss of generality, assume the hyper-rectangle starts at an odd point in the 1st dimension. Then:

$$\begin{aligned} b(2x_1+1, s_1, x_2, s_2, ..., x_n, s_n) &= b(2x_1+1, 1, x_2, s_2, ..., x_n, s_n) \\ &+ b(2x_1+2, s_1-1, x_2, s_2, ..., x_n, s_n) \end{aligned}$$

Clearly, the same principle can be used if the hyper-rectangle **ends** at an odd point. Figure 2 illustrates the slicing principle for a 2-dimensional space.

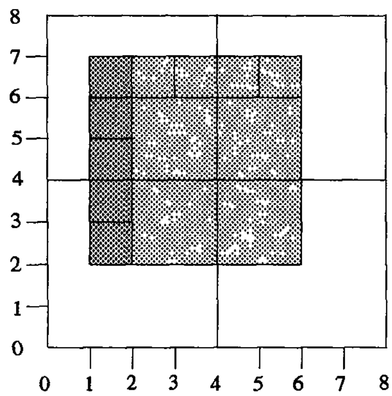

Figure 2: Slicing from the left, when the rectangle starts at an odd point (the left slice is more heavily shaded).

Observation 2 - 'Unit'. If any one dimension of a hyper-rectangle is of unit size, then it can be covered only with unit size blocks. Thus, the number of blocks required to cover it is equal to its volume and is obtained as the product of the sides, independent of position. That is:

$$b(x_1, s_1, x_2, s_2, ..., x_m, 1, ..., x_n, s_n) = \prod_{i=1}^{n} s_i$$

Observation 3 - 'Shrinking'. If a hyper-rectangle starts and ends at even numbers in all dimensions, then we can make the granularity coarser, maintaining the same number of blocks:

$$b(2x_1, 2s_1, 2x_2, 2s_2, ..., 2x_n, 2s_n) = b(x_1, s_1, x_2, s_2, ..., x_n, s_n)$$

Figure 3 gives a 2-dimensional example of the idea.

3 Solution for magic hyper-rectangles

Definition 2. A rectangle is called **magic** iff each side s_i is of the form $2^{m_i} - 1$.

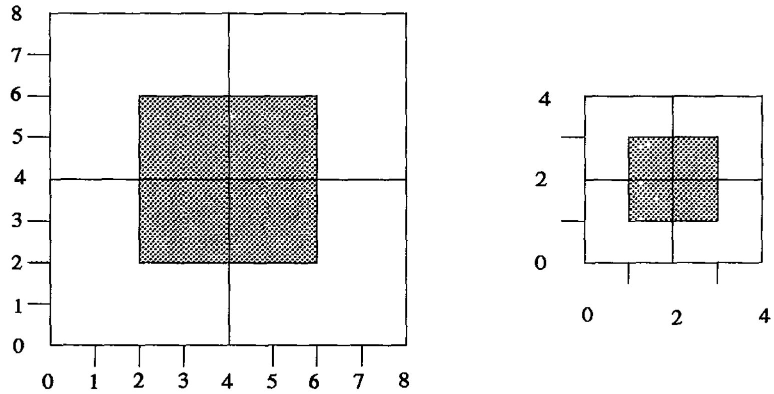

Figure 3: Halving the granularity.

Lemma 1. ('magic hyper-rectangles') If a rectangle is magic, then the number of blocks it decomposes to is **independent** of the position of the anchor:

$$b(x_1, 2^{m_1}-1, x_2, 2^{m_2}-1, ..., x_n, 2^{m_n}-1) \;=\; \text{constant} \qquad \forall (x_1, x_2, ..., x_n)$$

Proof. Without loss of generality, let s_1 be the smallest side of the hyper-rectangle. For every dimension i, we can apply the Slicing Observation exactly once, because every side s_i is odd. After that, all the sides are even, and the anchor points are even as well. So we can apply the Shrinking Observation; the resulting rectangle will still be magic: for every dimension i, after slicing and shrinking we will have a side of size: $(s_i - 1)/2 = (2^{m_i} - 1 - 1)/2 = 2^{m_i - 1} - 1$. Applying this step inductively, and using the Unit Observation as the base case, we have the required lemma. □

Corollary 1. For magic hyper-rectangles, we have:

$$\bar{b}(s_1, s_2, ..., s_n) \;=\; b(x_1, s_1, x_2, s_2, ..., x_n, s_n) \qquad \forall (x_1, x_2, ..., x_n)$$

□

Based on the last observation, we can quickly derive formulae for magic rectangles, bypassing equation (1).

3.1 Solution for magic hyper-cubes

Consider first a magic hyper-rectangle with all its sides the same size, that is, a hyper-cube. Let this size be $2^m - 1$.

Lemma 2. For a magic hyper-cube the number of blocks is:

$$\bar{b}(2^m - 1, ..., 2^m - 1) \;=\; (2^m - 1)^n - (2^n - 1) \sum_{t=1}^{m-1} (2^t - 1)^n$$

Proof. Independent of the position of the anchor, we 'slice off' one slice in each dimension, and then shrink. Thus:

$$b(x_1, 2^m - 1, ..., x_n, 2^m - 1) \;=\; (2^m - 1)^n - (2^m - 2)^n \;+\; \bar{b}(2^{m-1} - 1, ..., 2^{m-1} - 1) \qquad (2)$$

where the first two terms give the number of blocks contained in the slices, and the last term calculates the number of internal blocks. Solving this recursive relation (2) we have:

$$\bar{b}(2^m - 1, ..., 2^m - 1) \;=\; \sum_{t=1}^{m} \Big((2^t - 1)^n - (2^t - 2)^n\Big) \qquad (3)$$

or

$$\bar{b}(2^m - 1, ..., 2^m - 1) \;=\; (2^m - 1)^n - (2^n - 1) \sum_{t=1}^{m-1} (2^t - 1)^n$$

□

Next we try to find an approximation for large values of m.

Corollary 2. For a magic hyper-cube, the number of blocks is approximated by half of the hyper-surface S, if the side is large ($m \gg 1$) and the dimensionality is high ($n \gg 1$):

$$\bar{b}(2^m - 1, ..., 2^m - 1) \;\approx\; n\; 2^{m(n-1)} \;\approx S/2$$

Proof. Since we have $2^m - 1 \approx 2^m$, it follows that:

$$\begin{aligned}(2^t - 1)^n - (2^t - 2)^n \;&\approx\; n(2^t - 1)^{n-1} \\ &\approx\; n 2^{t(n-1)} \qquad (4)\end{aligned}$$

and, from equation (3) we obtain:

$$\begin{aligned}\bar{b}(2^m - 1, ..., 2^m - 1) \;&\approx\; \sum_{t=1}^{m} n\; \left(2^{(n-1)}\right)^t \\ &=\; \frac{n\; 2^{n-1}\; (2^{(n-1)m} - 1)}{2^{n-1} - 1} \qquad (5) \\ &\approx\; n\; 2^{m(n-1)} \qquad (6)\end{aligned}$$

Since the hyper-surface is given by:

$$\begin{aligned}S \;&=\; 2\; n\; (2^m - 1)^{(n-1)} \\ &\approx\; 2\; n\; 2^{m(n-1)}\end{aligned}$$

we have that, for large m and n:

$$\bar{b}(2^m - 1, ..., 2^m - 1) \;\approx\; S/2 \qquad (7)$$

which says that the number of quadtree blocks is approximately half of the hyper-surface. □

The above holds if $n \gg 1$. For 2-d space, which is of much interest, we obtain, from Eq. 5 with $n = 2$:

$$\bar{b}(2^m - 1, 2^m - 1) \approx 2 \times 2 \times (2^m - 1) = S$$

which agrees with the result of Hunter and Steiglitz [13], stating that the number of quadtree nodes for a polygon is proportional to its perimeter.

3.2 Extension to any magic hyper-rectangle

For a magic hyper-rectangle, without loss of generality, let $s_1 = 2^m - 1$ be its smallest side. Also, let $s_i = 2^{m+d_i} - 1$ where $d_i \geq 0$. In other words, we assume that: $d_1 = 0$.

Lemma 3. For any magic hyper-rectangle the number of blocks is:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1, ..., 2^{m+d_n} - 1) = \prod_{i=1}^{n}(2^{m+d_i} - 1) - (2^n - 1)\sum_{j=1}^{m-2}\prod_{i=1}^{n}(2^{m-j+d_i} - 1)$$

Proof. Using the Slicing and Shrinking Observations as we did for the magic hyper-cubes, we have:

$$\begin{aligned}\bar{b}(2^m - 1, 2^{m+d_2} - 1, ..., 2^{m+d_n} - 1) &= (2^m - 1)(2^{m+d_2} - 1)...(2^{m+d_n} - 1) \\ &- (2^m - 2)(2^{m+d_2} - 2)...(2^{m+d_n} - 2) \\ &+ \bar{b}(2^{m-1} - 1, 2^{m-1+d_2} - 1, ..., 2^{m-1+d_n} - 1)\end{aligned}$$

Solving the recursion (it bottoms after m steps), we have:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1, ..., 2^{m+d_n} - 1) = \sum_{t=1}^{m}\left(\prod_{i=1}^{n}(2^{t+d_i} - 1) - \prod_{i=1}^{n}(2^{t+d_i} - 2)\right) \quad (8)$$

or

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1, ..., 2^{m+d_n} - 1) = \prod_{i=1}^{n}(2^{m+d_i} - 1) - (2^n - 1)\sum_{j=1}^{m-2}\prod_{i=1}^{n}(2^{m-j+d_i} - 1)$$

□

Again, we try to find an approximation for large m.

Corollary 3. Equation (8) can be approximated by:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1, ..., 2^{m+d_n} - 1) \approx 2^{m(n-1)} \sum_{j=2}^{n} 2^{-d_j} \prod_{i=2}^{n} 2^{d_i}$$

Proof. By using a reasoning similar to that of equation (4) we have:

$$\begin{aligned}&\prod_{i=1}^{n}(2^{t+d_i} - 1) - \prod_{i=1}^{n}(2^{t+d_i} - 2) \\ &= (2^t - 1)(2^{t+d_2} - 1)...(2^{t+d_n} - 1) - (2^t - 2)(2^{t+d_2} - 2)...(2^{t+d_n} - 2) \\ &\approx (2^t)^{n-1} \sum_{j=2}^{n} 2^{-d_j} \prod_{i=2}^{n} 2^{d_i}\end{aligned}$$

Thus, by using the approximation of equation (6), equation (8) becomes:

$$\begin{aligned}\bar{b}(2^m-1, 2^{m+d_2}-1, ..., 2^{m+d_n}-1) &\approx \sum_{t=1}^{m} (2^t)^{n-1} \sum_{j=2}^{n} 2^{-d_j} \prod_{i=2}^{n} 2^{d_i} \\ &= \sum_{j=2}^{n} 2^{-d_j} \prod_{i=2}^{n} 2^{d_i} \sum_{t=1}^{m} (2^t)^{n-1} \\ &\approx 2^{m(n-1)} \sum_{j=2}^{n} 2^{-d_j} \prod_{i=2}^{n} 2^{d_i}\end{aligned}$$

and we conclude once more that for high dimensionalities n and large hyper-rectangles ($m \gg 1$), $\bar{b}()$ is roughly half of the hyper-surface. □

4 Proof of linearity

In the previous section we solved the problem for magic hyper-rectangles. Here we show how to solve the problem for arbitrary rectangles using *linear interpolation.*

Lemma 4. If $x_1 + s_1$ is odd, then:

$$b(x_1, s_1, x_2, s_2, ..., x_n, s_n) = b(x_1, s_1 - 1, x_2, s_2, ..., x_n, s_n) + C_1$$

where C_1 is a constant independent of the specific values of x_1 and s_1.

Proof. The hyper-cubes to cover the incremental volume are forced to be no more than 1 unit in the first dimension, and therefore 1 unit in each dimension. The number of hyper-cubes required is simply $s_2 \times s_3 \times ... \times s_n$, by following the Unit Observation. Define C_1 to be $\prod_{i=2}^{n} s_i$ to complete the proof. □

Lemma 5. If $x_1 + s_1$ is even, but not divisible by 4, then:

$$b(x_1, s_1, x_2, s_2, ..., x_n, s_n) = b(x_1, s_1 - 1, x_2, s_2, ..., x_n, s_n) + C_2$$

where C_2 is a constant independent of the specific values of x_1 and s_1.

Proof. Now, some of the hyper-cubes already used to cover the hyper-rectangle may be merged with the new layer added into larger blocks, 2 units on the side, on the even boundaries. The number of such mergers possible is determined solely by the size and position in dimensions $2, ..., n$ and is independent of x_1 and s_1. Call the number of additional blocks required C_2. □

Lemma 6. If $x_1 + s_1$ is divisible by 2^{j-1} but not by 2^j, and $s_1 \geq 2^{j-1}$ then:

$$b(x_1, s_1, x_2, s_2, ..., x_n, s_n) = b(x_1, s_1 - 1, x_2, s_2, ..., x_n, s_n) + C_j$$

where C_j is a constant independent of the specific values of x_1 and s_1.

Proof. Similar to Lemma 5. The additional condition imposing a minimum limit on s_1 is required since clearly no more mergers are possible beyond the length of the side s_1. Yet, the construction in the lemma could require mergers into blocks up to 2^{j-1} on the side. □

Lemma 7. If $x_1 + s_1$ is divisible by 2^j and $2^{m-1} \leq s_1 < 2^m \leq 2^j$, then:

$$b(x_1, s_1, x_2, s_2, ..., x_n, s_n) \;=\; b(x_1, s_1 - 1, x_2, s_2, ..., x_n, s_n) \;+\; C_m$$

where C_m is a constant independent of the specific values of x_1 and s_1.

Proof. Similar to Lemma 6. Since s_1 is too small, the merger of blocks cannot continue until a side of 2^j is reached. Instead, it stops at an earlier point, and this point is determined by the magic points between which s_1 lies but is otherwise independent of s_1 and x_1. □

Now we are in the position to state the main theorems.

Theorem 1. For an arbitrary hyper-rectangle with sides $(s_1, s_2, ..., s_n)$, where $2^{m-1} \leq s_1 < 2^m - 1$ we have:

$$\bar{b}(s_1, s_2, ..., s_n) - \bar{b}(s_1 - 1, s_2, ..., s_n) \;=\; \bar{b}(s_1 + 1, s_2, ..., s_n) - \bar{b}(s_1, s_2, ..., s_n)$$

Proof. Consider the expected number of hyper-cube blocks to cover a hyper-rectangle $\bar{b}(s_1 - 1, s_2, ..., s_n)$. If $s_1 - 1$ is increased to s_1, then following the lemmas above, the increase in the value $\bar{b}()$ is independent of the specific value of s_1, as long as a magic threshold is not crossed. Since the value of x_1 is arbitrary, independent of the specific value of s_1 we have that $x_1 + s_1$ is divisible by 2 with probability 1/2, by 4 with probability 1/4, and so on. Therefore the number of additional blocks required is C_1 with probability 1/2, C_2 with probability $1/2^2$, and so C_j with probability $1/2^j$, until C_m with probability $1/2^m$ and C_{m+1} with probability $1/2^m$. Thus, all cases are taken in consideration and their respective probabilities sum to unity. Note, also, that divisibility by higher powers of 2 does not alter the constant, and hence we can sum all these terms into a single term. Call this summation C:

$$C \;=\; C_1/2 + C_2/4 + \ldots + C_m/2^m + C_{m+1}/2^m \tag{9}$$

Exactly the same summation C is obtained if s_1 is now increased to $s_1 + 1$. Thus the theorem is established. □

Theorem 2. Let $R = s_1 \times s_2 ... \times s_n$ be a hyper-rectangle; let m_1 and M_1 be the magic values that contain s_1 (i.e., $m_1 = 2^j - 1 \leq s_1 < 2^{j+1} - 1 = M_1$), with similar definitions for m_i and M_i. There are 2^n magic rectangles that we can generate (for each dimension i, we have two choices: m_i and M_i, for a total of 2^n choices). The number of blocks for R is determined by a linear interpolation among the values of the above 2^n magic rectangles.

Proof. Consider each dimension in turn and increase the size from m_i to M_i in steps of 1. Each step increases the expected number of blocks by the same amount, on account of Theorem 1. While Theorem 1 was established for the 1st dimension, by arguments of symmetry it holds for all other dimensions as well. Therefore, the increase from m_i to s_i is a linear interpolation of the increase from m_i to M_i. The order in which the dimensions are considered is immaterial. □

In other words, the function $\bar{b}(s_1 - 1, s_2, ..., s_n)$ is *piece-wise linear* on its arguments, with 'break points' whenever a value s_i is a magic number. Table 2 shows the values for $\bar{b}()$ for the 2-dimensional case, with boldface numbers for the magic rectangles. Notice that the rest of the numbers can be derived by linear interpolation among the 4 magic rectangles nearest to the point of interest. (e.g., for the $\bar{b}(5,2)$, the corresponding magic rectangles are (3,1), (3,3), (7,1), (7,3)). In the next section we illustrate the Theorem 2, deriving the formulae for $\bar{b}()$ for 2-dimensional and 3-dimensional spaces.

5 Examples: 2- and 3-dimensional rectangles

In this section we illustrate the steps of the lemmas and theorems of the previous section by deriving closed-form exact formulae for the expected number of blocks a 2-dimensional and a 3-dimensional rectangle. Following the steps of the previous section, we first calculate the number of blocks for any magic rectangular object, and then we give exact formulae for any (non-magic) rectangular object.

5.1 2-dimensional rectangles

This case has been analyzed in [8]. Here, we show how those results can be derived as special cases of the Theorems and Lemmas of the previous section.

Lemma 8. The average number of blocks $\bar{b}()$ that a magic rectangle in 2-dimensional space decomposes into is:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1) \;=\; 2\,(2^m - 1)\,(2^{d_2} + 1) \;-\; 3m \tag{10}$$

Proof. From expression (8) we have:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1) \;=\; \sum_{t=1}^{m} \left(\prod_{i=1}^{2} (2^{t+d_i} - 1) - \prod_{i=1}^{2} (2^{t+d_i} - 2) \right) \tag{11}$$

It is sufficient to prove that the right hand parts of relations (10) and (11) are equal. The proof follows by induction on m. For $m = 1$ both sides of the equation are equal to: $2^{d_2+1} - 1$. For $m = 2$ both sides are equal to: $3 * 2^{d_2+1}$. We assume that the above relation holds for $m = k$:

$$\sum_{t=1}^{k} \left(\prod_{i=1}^{2} (2^{t+d_i} - 1) - \prod_{i=1}^{2} (2^{t+d_i} - 2) \right) \;=\; 2\,(2^k - 1)\,(2^{d_2} + 1) \;-\; 3k$$

We will prove that it holds for $m = k + 1$:

$$\sum_{t=1}^{k+1} \left(\prod_{i=1}^{2} (2^{t+d_i} - 1) - \prod_{i=1}^{2} (2^{t+d_i} - 2) \right) \;=\; 2\,(2^{k+1} - 1)\,(2^{d_2} + 1) \;-\; 3(k + 1)$$

It is sufficient to prove that the left hand part of the above equation is:

$$\begin{aligned}
&2\,(2^{k+1} - 1)\,(2^{d_2} + 1) \;-\; 3(k + 1) \;= \\
&2\,(2^k - 1)\,(2^{d_2} + 1) \;-\; 3k \;+\; (2^{k+1} - 1)\,(2^{k+1+d_2} - 1) \;-\; (2^{k+1} - 2)\,(2^{k+1+d_2} - 2)
\end{aligned}$$

After some simple algebra we derive that the above lemma holds. □

Table 2 gives the number of blocks a rectangle is decomposed into, when its sides s_1 and s_2 are smaller than 9. The entries were calculated by exhaustive enumeration, using the definition of Eq. 1. Entries corresponding to magic rectangles are in boldface. In the sequel, we will show how the remaining entries have been filled. Next, we trace the steps of the proof of Theorem 1, giving a closed formula for the constant C.

s_2 / s_1	1	2	3	4	5	6	7	8
1	**1**	2	**3**	4	5	6	**7**	8
2	2	3.25	4.5	5.75	7	8.25	9.5	10.75
3	**3**	4.5	**6**	7.5	9	10.5	**12**	13.5
4	4	5.75	7.5	9.0625	10.625	12.1875	13.75	15.3125
5	5	7	9	10.625	12.25	13.875	15.5	17.125
6	6	8.25	10.5	12.1875	13.875	15.5625	17.25	18.9375
7	**7**	9.5	**12**	13.75	15.5	17.25	**19**	20.75
8	8	10.75	13.5	15.3125	17.125	18.9375	20.75	22.515625

Table 2: Number of blocks for 2-dimensional rectangles. Magic rectangles are in boldface.

Lemma 9. Given that the rectangle with sides (s_1, s_2) is magic, then the number of blocks for a rectangle with sides $(s_1 + 1, s_2)$ is:

$$\bar{b}(s_1+1, s_2) \; = \; \bar{b}(s_1, s_2) \; + \; 2^{m+d_2-2max} \; - \; 3 * 2^{-max} \; + \; 2$$

where $max = \lfloor \log(\min(s_1+1, s_2)) \rfloor$.

Proof. See appendix A. □

It is evident that in a 2-dimensional space the constant C of Theorem 1 is given by:

$$C = 2^{m+d_2-2max} - 3 * 2^{-max} + 2$$

We can rewrite this expression as: $(s_2 - 1) * 2^{-2max} - 3 * 2^{-max} + 2$ from which we can see that this constant C is independent of x_1, s_1. In the following corollary we will use the symbol $C(s_i)$ to denote this function of the quantity s_i. Thus:

$$C(s_i) \; = \; (s_i - 1) * 2^{-2max} \; - \; 3 * 2^{-max} \; + \; 2$$

5.2 3-dimensional rectangles

In this subsection, we examine the case of a parallelepiped and we derive a formula for the constant C of Theorem 2.

Lemma 10. The number of blocks that a magic parallelepiped decomposes into is:

$$\bar{b}(2^m - 1, 2^{m+d_2} - 1, 2^{m+d_3} - 1) = \sum_{t=1}^{m} \left(\prod_{i=1}^{3} (2^{t+d_i} - 1) - \prod_{i=1}^{3} (2^{t+d_i} - 2) \right)$$

$$= \frac{4}{3}(2^{2m} - 1)(2^{d_2} + 2^{d_3} + 2^{d_2+d_3}) - 6(2^m - 1)(1 + 2^{d_2} + 2^{d_3}) + 7$$

Proof. By induction on m. □

Lemma 11. Given that 3-dimensional parallelepiped with sides (s_1, s_2, s_3) is magic, then the number of blocks for a parallelepiped with sides $(s_1 + 1, s_2, s_3)$ is:

$$\bar{b}(s_1 + 1, s_2, s_3) = \bar{b}(s_1, s_2, s_3) + 2^{2m+d_2+d_3-3max} + 8 - \frac{7}{3}\left(2^{max+1} + 2^{-max}\right) - 2^m \left(2^{d_2} + 2^{d_3}\right) \left(\frac{7}{9} 2^{-2max} - \frac{7}{6} max + \frac{2}{9}\right)$$

where $max = \lfloor \log(\min(s_1 + 1, s_2, s_3)) \rfloor$.

Proof. See appendix B.

From Lemma 11 we understand why the constant C of Theorem 1 is a quantity independent of s_1. However, we observe that it depends on the other two sides s_2 and s_3. This is the reason why for the case of a 3-dimensional space we have to denote this quantity as $C(s_i, s_j)$, where:

$$C(s_i, s_j) = (s_i - 1) * (s_j - 1) * 2^{-3max} + (s_i + s_j - 2) * \left(\frac{7}{9} 2^{-2max} - \frac{7}{6} max + \frac{2}{9}\right) - \frac{7}{3}(2^{max+1} + 2^{-max}) + 8 \qquad (12)$$

Table 3 gives the number of blocks a parallelepiped is composed of, when its sides are smaller than 6. Entries in boldface correspond to magic parallelepipeds. All the entries have been computed using exhaustive enumeration, from the definition of Eq. 1.

s_1 \ s_2	1	2	3	4	5	s_3
	1	2	**3**	4	5	1
	2	4	6	8	10	2
1	**3**	6	**9**	12	15	3
	4	8	12	16	20	4
	5	10	15	20	25	5
	2	4	6	8	10	1
	4	7.125	10.25	13.375	16.5	2
2	6	10.25	14.5	18.75	23	3
	8	13.375	18.75	24.125	29.5	4
	4	16.5	23	29.5	36	5
	3	6	**9**	12	15	1
	6	10.25	14.5	18.75	23	2
3	**9**	14.5	**20**	25.5	31	3
	12	18.75	25.5	32.25	39	4
	15	23	31	39	47	5
	4	8	12	16	20	1
	8	13.375	18.75	24.125	29.5	2
4	12	18.75	25.5	32.25	39	3
	16	24.125	32.25	40.265625	48.28125	4
	20	29.5	39	48.28125	57.5625	5
	5	10	15	20	25	1
	10	16.5	23	29.5	36	2
5	15	23	31	39	47	3
	20	29.5	39	48.28125	57.5625	4
	25	36	47	57.5625	68.125	5

Table 3: Number of blocks for 3-dimensional parallelepipeds. Magic parallelepipeds are in boldface.

6 Discussion and conclusions

We have examined the problem of the number of quad-tree blocks that an n-dimensional rectangle will be decomposed into on the average. There are two interesting observations:

- Our approach (Theorem 2 and Eq. 8) generalizes all the older approaches on 2-dimensional rectangles [6, 8, 21].
- It generalizes the observation of Hunter and Steiglitz [13] that the number of quadtree blocks is proportional to the perimeter of the polygon. Our formula shows that, for 2-dimensional rectangles, the number of quadtree blocks is approximately the perimeter of the rectangle, while for higher dimensionalities $n \gg 1$, it is roughly half of the hyper-surface.

The contributions of this paper are both practical and theoretical. From the practical point of view, the number of quadtree blocks of a decomposition is important, because it determines the number of nodes that a main-memory-based quadtree will require; the number of entries in a linear quadtree that will be required; also, the number of pieces that a range query will be decomposed into (which will be proportional to the response time for this query).

From the theoretical point of view, it proposes a methodology which we believe will be useful in the analysis of other quadtree-related methods (e.g., methods using space-filling curves, such as the z-ordering [17], Gray codes [7], or the Hilbert curve [10]). The methodology consists of two steps:

Step 1 solve the problem for the 'magic' rectangles (which is easy)

Step 2 show that the formula for an arbitrary rectangle can be derived by linear interpolation from suitable 'magic' rectangles.

Future work includes the extension of this method for the analysis of rectilinear polygons (including concave ones), as well as the analysis for space filling curves for 2-dimensional and n-dimensional spaces.

A Appendix: Lemma for the 2-dimensional case

Lemma 9. Given that the rectangle with sides (s_1, s_2) is magic, then the number of blocks for a rectangle with sides (s_1+1, s_2) is:

$$\bar{b}(s_1+1, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; 2^{m+d_2-2max} \;-\; 3 * 2^{-max} \;+\; 2$$

where $max = \lfloor \log(\min(s_1+1, s_2)) \rfloor$.

Proof. First, let's assume that the rectangle does not wrap around the edges $(x_1+s_1, x_2+s_2 \leq K)$. With probability 1/2 we have: $(x_1+s_1+1) \bmod 2 \neq 0$ (the end point in the 1st dimension is an odd number). Then according to the Slicing and Unit Observations the new number of blocks is:

$$b(x_1, s_1+1, x_2, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; s_2$$

With probability equal to 1/4 we have: $(x_1+s_1+1) \bmod 2 = 0$ but $(x_1+s_1+1) \bmod 4 \neq 0$. Then:

$$b(x_1, s_1+1, x_2, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \left(\lfloor\frac{x_2+s_2}{2}\rfloor - \lceil\frac{x_2}{2}\rceil\right)\,(2^1+2^1-1) \qquad (13)$$

The product in the previous relation stands for the number of blocks we have to subtract because mergings have been performed. The first two terms in the second parenthesis respectively stand for the number of pixels of the original magic rectangle (2^1) and for the number of the pixels of the additional slice (2^1) that merge in one 2x2 block. Thus, the third term in the parenthesis (i.e., -1) stands for the greater formed block we have to take into account. The first parenthesis of the product gives the number of greater blocks that may be formed.

Since s_2 is an odd integer (of the form $2^{m+d_2}-1$), it is easily verifiable that:

$$\lfloor\frac{x_2+s_2}{2}\rfloor - \lceil\frac{x_2}{2}\rceil \;=\; \lfloor\frac{s_2}{2}\rfloor$$

Thus, relation (13) becomes:

$$b(x_1, s_1+1, x_2, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \lfloor\frac{s_2}{2}\rfloor\,(2^1+2^1-1) \qquad (14)$$

With probability equal to 1/8 we have: $(x_1+s_1+1) \bmod 4 = 0$ but $(x_1+s_1+1) \bmod 8 \neq 0$. Then:

$$\begin{aligned} b(x_1, s_1+1, x_2, s_2) \;&=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \left(\lfloor\frac{x_2+s_2}{4}\rfloor - \lceil\frac{x_2}{4}\rceil\right)\,(2^1+2^2+2^2-1) \;- \\ &\quad \lfloor\frac{s_2 - 4(\lfloor\frac{x_2+s_2}{4}\rfloor - \lceil\frac{x_2}{4}\rceil)}{2}\rfloor\,(2^1+2^1-1) \;\Rightarrow \\ b(x_1, s_1+1, x_2, s_2) \;&=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \lfloor\frac{s_2}{4}\rfloor\,(2^1+2^2+2^2-1) \;- \\ &\quad \lfloor\frac{s_2 - 4\lfloor\frac{s_2}{4}\rfloor}{2}\rfloor\,(2^1+2^1-1) \end{aligned}$$

Since: $\sum_{j=1}^{i} 2^j \;=\; 2(2^i-1)$, the above relation becomes:

$$b(x_1, s_1+1, x_2, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \lfloor\frac{s_2}{4}\rfloor\;3\;(2^2-1) \;-\; \lfloor\frac{s_2 \bmod 4}{2}\rfloor\;3\;(2^1-1)$$

Suppose that: $8 \leq \min(s_1+1, s_2) < 16$. Then with probability equal to 1/8 we have: $(x_1+s_1+1) \bmod 4 = 0$ and $(x_1+s_1+1) \bmod 8 = 0$. Thus:

$$\begin{aligned} b(x_1, s_1+1, x_2, s_2) \;&=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \lfloor\frac{s_2}{8}\rfloor\;3\;(2^3-1) \;-\; \lfloor\frac{s_2 \bmod 8}{4}\rfloor\;3\;(2^2-1) \;- \\ &\quad \lfloor\frac{s_2 \bmod 4}{2}\rfloor\;3\;(2^1-1) \end{aligned}$$

Following this reasoning, similar expressions can be derived for large values of s_1, s_2 and such that $K/2 < \min(s_1+1, s_2) \leq K, \forall K = 2^k$.

Secondly, suppose that the rectangle wraps around in one dimension only (i.e. $x_2 + s_2 > K$). Then, expression (13) should be rewritten as:

$$b(x_1, s_1+1, x_2, s_2) \;=\; \bar{b}(s_1, s_2) \;+\; s_2 \;-\; \left(\lfloor\frac{x_2+s_2-K}{2}\rfloor + \lfloor\frac{K-x_2}{2}\rfloor\right)\;(2^1+2^1-1)$$

However, the latter expression may be reduced to (14). This way, the set of equations derived by assuming that the rectangle wraps around only one edge reduces to the set of equations produced to describe the no-wrapping rectangle. The same result holds even if the rectangle wraps around both edges.

Thus, by considering all the positions possibly taken by the end point in the 1st dimension, we conclude to the following expression:

$$\begin{aligned}\bar{b}(s_1+1, s_2) \;=\;& \bar{b}(s_1, s_2) \;+\; s_2 \;-\\ & \sum_{i=1}^{max-1} \frac{1}{2^{i+1}} \left(\lfloor\frac{s_2}{2^i}\rfloor \; 3\;(2^i-1) \;+\; \sum_{j=2}^{i} \lfloor\frac{s_2 \bmod 2^j}{2^{j-1}}\rfloor \; 3\;(2^{j-1}-1)\right) \;-\\ & \frac{1}{2^{max}} \left(\lfloor\frac{s_2}{2^{max}}\rfloor \; 3\;(2^{max}-1) \;+\; \sum_{j=2}^{max} \frac{s_2 \bmod 2^j}{2^{j-1}} \; 3\;(2^{j-1}-1)\right)\end{aligned}$$

which is averaged and independent of the anchor point (x_1, x_2). Since: $s_2 = 2^{m+d_2}-1$, the floor functions are simplified to unity and after some algebra on geometric series the lemma is proved. Notice, also, that if $d_2 > 0$ then $max = log(s_1+1) = m$, whereas if $d_2 = 0$ then $max = log(s_2) = m-1$. □

B Appendix: Lemma for the 3-dimensional case

Lemma 11. Given that 3-dimensional parallelepiped with sides (s_1, s_2, s_3) is magic, then the number of blocks for a parallelepiped with sides (s_1+1, s_2, s_3) is:

$$\begin{aligned}\bar{b}(s_1+1, s_2, s_3) \;=\;& \bar{b}(s_1, s_2, s_3) \;+\; 2^{2m+d_2+d_3-3max} \;+\; 8 \;-\; \frac{7}{3}\,(2^{max+1} \;+\; 2^{-max}) \;-\\ & 2^m\;(2^{d_2}+2^{d_3})\;\left(\frac{7}{9}2^{-2max} - \frac{7}{6}max + \frac{2}{9}\right)\end{aligned}$$

where $max = \lfloor\log(\min(s_1+1, s_2, s_3))\rfloor$.

Proof. We follow the same reasoning as for the case of Lemma 9. If $(x_1+s_1+1) \bmod 2 \neq 0$ (which may happen with probability 1/2), then according to the Slicing and Unit Observations we calculate the new number of blocks to be:

$$b(x_1, s_1+1, x_2, s_2, x_3, s_3) \;=\; \bar{b}(s_1, s_2, s_3) \;+\; s_2 * s_3$$

If $(x_1 + s_1 + 1) \bmod 2 = 0$ but $(x_1 + s_1 + 1) \bmod 4 \neq 0$, then with probability equal to 1/4 we have:

$$b(x_1, s_1+1, x_2, s_2, x_3, s_3) = \bar{b}(s_1, s_2, s_3) + s_2 * s_3 - \lfloor \frac{s_2}{2} \rfloor \lfloor \frac{s_3}{2} \rfloor \left((2^1)^2 + (2^1)^2 - 1\right)$$

In an analogous manner, with probability equal to 1/8 (for the case $(x_1 + s_1 + 1) \bmod 4 = 0$ but $(x_1 + s_1 + 1) \bmod 8 \neq 0$), we have:

$$\begin{aligned}
&b(x_1, s_1+1, x_2, s_2, x_3, s_3) \\
&\quad = \bar{b}(s_1, s_2, s_3) + s_2 * s_3 - \lfloor \frac{s_2}{4} \rfloor \lfloor \frac{s_3}{4} \rfloor \left((2^1)^2 + (2^2)^2 + (2^2)^2 - 1\right) - \left((2^1)^2 + (2^1)^2 - 1\right) \\
&\quad = \bar{b}(s_1, s_2, s_3) + s_2 * s_3 - \lfloor \frac{s_2}{4} \rfloor \lfloor \frac{s_3}{4} \rfloor \left(\frac{7}{3}(4^2 - 1)\right) - \left(\frac{7}{3}(4^1 - 1)\right)
\end{aligned}$$

Thus, by generalizing and considering all the positions possibly taken by the end point in the 1st dimension, we conclude to the following expression:

$$\begin{aligned}
\bar{b}(s_1+1, s_2, s_3) = {} & \bar{b}(s_1, s_2, s_3) + s_2 * s_3 - \\
& \sum_{i=1}^{max-1} \frac{1}{2^{i+1}} \left(\lfloor \frac{s_2}{2^i} \rfloor \lfloor \frac{s_3}{2^i} \rfloor \frac{7}{3}(4^i - 1) + \sum_{j=2}^{i} \frac{7}{3}(4^{j-1} - 1) \right) - \\
& \frac{1}{2^{max}} \left(\lfloor \frac{s_2}{2^{max}} \rfloor \lfloor \frac{s_3}{2^{max}} \rfloor \frac{7}{3}(4^{max} - 1) + \sum_{j=2}^{max} \frac{7}{3}(4^{j-1} - 1) \right)
\end{aligned}$$

which is averaged and independent of the anchor point (x_1, x_2, x_3). After some algebra the expression of the lemma follows. □

References

[1] Walid G. Aref and Hanan Samet. Optimization strategies for spatial query processing. *Proc. of VLDB (Very Large Data Bases)*, pages 81–90, September 1991.

[2] Manish Arya, William Cody, Christos Faloutsos, Joel Richardson, and Arthur Toga. Qbism: Extending a dbms to support 3d medical images. *Tenth Int. Conf. on Data Engineering (ICDE)*, pages 314–325, February 1994.

[3] D. Ballard and C. Brown. *Computer Vision.* Prentice Hall, 1982.

[4] J.L. Bentley. Multidimensional binary search trees used for associative searching. *CACM*, 18(9):509–517, September 1975.

[5] Mathematical Committee on Physical and NSF Engineering Sciences. *Grand Challenges: High Performance Computing and Communications.* National Science Foundation, 1992. The FY 1992 U.S. Research and Development Program.

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

[7] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Trans. on Software Engineering*, 14(10):1381–1393, October 1988. early version available as UMIACS-TR-87-4, also CS-TR-1796.

[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 Y. Rong. Dot: a spatial access method using fractals. In *IEEE Data Engineering Conference*, pages 152–159, Kobe, Japan, April 1991. early version available as UMIACS-TR-89-31, CS-TR-2214.

[10] 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. also available as UMIACS-TR-89-47 and CS-TR-2242.

[11] R.A. Finkel and J.L. Bentley. Quadtrees: a data structure for retrieval on composite keys. *ACTA Informatica*, 4(1):1–9, 1974.

[12] K. Hinrichs and J. Nievergelt. The grid file: a data structure to support proximity queries on spatial objects. *Proc. of the WG'83 (Intern. Workshop on Graph Theoretic Concepts in Computer Science)*, pages 100–113, 1983.

[13] G.M. Hunter and K. Steiglitz. Operations on images using quad trees. *IEEE Trans. on PAMI*, PAMI-1(2):145–153, April 1979.

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

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

[16] Curtis P. Kolovson and Michael Stonebraker. Segment indexes: Dynamic indexing techniques for multi-dimensional interval data. *Proc. ACM SIGMOD*, pages 138–147, May 1991.

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

[18] J.A. Orenstein and F.A. Manola. Probe spatial data modeling and query processing in an image database application. *IEEE Trans. on Software Engineering*, 14(5):611–629, May 1988.

[19] 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.

[20] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, 1989.

[21] 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.

[22] C.A. Shaffer, H. Samet, and R.C. Nelson. Quilt: a geographic information system based on quadtrees. Technical Report CS-TR-1885.1, Univ. of Maryland, Dept. of Computer Science, July 1987. to appear in the International Journal of Geographic Information Systems.

[23] M. White. *N-Trees: Large Ordered Indexes for Multi-Dimensional Space.* Application Mathematics Research Staff, Statistical Research Division, U.S. Bureau of the Census, December 1981.