On the Average Performance of Orthogonal Range Search in Multidimensional Data Structures*

Amalia Duch† Conrado Martínez‡

November 23, 2001

Abstract

In this work we present the average-case analysis of orthogonal range search for several multidimensional data structures. We first consider random relaxed K-d trees as a prototypical example and later extend our results to several different multidimensional data structures. We show that the performance of range searches is related to the performance of a variant of partial matches which simplifies the analysis and provides a tight asymptotic estimate for the expected cost of range searches.

1 Introduction

Orthogonal range search appears frequently in applications of large databases, geographical information systems, multimedia databases and computer graphics, among others [14]. Given a collection of multidimensional data points and a query rectangle, the goal of an orthogonal range search (range search, for short) is to retrieve all the data points in the collection that fall inside the given rectangle. Besides this most direct application, range search is implicitly involved in more complex region queries and other associative queries.

Over the years, many data structures have been proposed for the management of multidimensional data and specifically for range search (see for instance [3]). Among these, K-d trees [1], quadtrees [2], K-d tries [13] and multiple variants of these.

*This research was supported by project DGES PB98-0926 (AEDRI) of the Spanish Ministry for Education and Science. The second author was also supported the Future and Emergent Technologies programme of the EU under contract IST-1999-14186 (ALCOM-FT).

†Laboratorio Nacional de Informática Avanzada (LANIA), Xalapa, Mexico. E-mail: amalia@xalapa.lania.mx.

‡Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain. E-mail: conrado@lsi.upc.es.

The mathematical analysis of the performance of range searches has proven a difficult task. Most previous analyses assumed that the tree data structure is balanced (e.g. [4, 15]), which often yields unduly optimistic results. Only recently there has been notable progress on this direction with two recent papers [5, 7] that provide upper and lower bounds for the average performance of range search in standard K-d trees, squarish K-d trees (a variant of the former introduced in [7]) and other multidimensional data structures.

In this work we analyze the average cost of range queries in randomized K-d trees [8] and many other multidimensional data structures. We begin reviewing random(ized) relaxed K-d trees and orthogonal range queries in Section 2, as well as the random models used in the sequel. In Section 3 we introduce *sliced partial matches* and relate the performance of range search with the performance of sliced partial matches; we use this relationship to provide a tight asymptotic analysis of the average cost of range search. While our results are analogous to those in [5, 7], the proof techniques are rather different and our techniques provide tight asymptotic estimates. In Section 4 we discuss the extension of the results for relaxed K-d trees to other multidimensional data structures: standard K-d trees, squarish K-d trees, K-d-t trees, standard and relaxed K-d tries, quadtrees and quadtries.

In the appendix at the end of this extended abstract we report the results of a preliminar experimental study.

2 Basic definitions

Let $F = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, $n \geq 0$ be the file of K-dimensional data points. We assume that each $x \in F$ is a K-tuple $x = (x_0, \ldots, x_{K-1})$ in $[0,1]^K$.

A *relaxed K-d tree* [8] for a set F of K-dimensional data points is a binary tree in which: (a) each node contains a K-dimensional data point and has associated a discriminant $j \in \{0, 1, \ldots, K-1\}$; (b) for each node x and discriminant j, the following invariant is true: any data point y in the left subtree satisfies $y_j < x_j$ and any data point z in the right subtree satisfies $z_j \geq x_j$.

Notice that the sequence of discriminants in a path from the root to any leaf is arbitrary. On the contrary, the definition of standard K-d trees [1] requires the sequence of discriminants along any path to be cyclic, starting with $j = 0$. Thus the root of the tree at level 0 discriminates with respect to the first coordinate ($j = 0$), its sons at level 1 discriminate w.r.t. the second coordinate ($j = 1$), and in general, all nodes at level m discriminate w.r.t. coordinate $j = m \mod K$.

Our average-case analysis of range searches over relaxed K-d trees in Section 3 will assume that trees are *random*. We say that a relaxed K-

d tree of size n is *random* if it is built by n insertions where the points are drawn independently from a continuous distribution in $[0,1]^K$ and the discriminants are drawn uniformly and independently from $\{0,\ldots,K-1\}$.

In random relaxed K-d trees, this assumption about the distribution of the input implies that the $n!^K K^n$ possible configurations of input file and discriminant sequences are equiprobable [8]. For standard K-d trees, since there is a fixed rule for discriminants, the assumption of random insertions (producing random standard K-d trees) implies that all $n!^K$ input sequences are equiprobable [1, 11]. In particular, in a random relaxed K-d tree each of the $n \cdot K$ possibilities of (key, discriminant) pairs are equally likely to appear in the root and once the root is fixed, the left and right subtrees are independent random relaxed K-d trees.

A *range query* is a K-dimensional hyperrectangle Q. We shall write $Q = [\ell_0, u_0] \times [\ell_1, u_1] \times \cdots \times [\ell_{K-1}, u_{K-1}]$, with $\ell_i \leq u_i$, for $0 \leq i < K$.

We will use the probabilistic model of random range queries introduced in [5, 7]. In this model, the center of a *random range query* Q is an independently drawn point z in $[0,1]^K$, sampled from the same continuous distribution as the data points, and the edges of the query have given lengths $\Delta_0, \Delta_1, \ldots \Delta_{K-1}$, with $0 \leq \Delta_i \leq 1$, for $0 \leq i < K$. Therefore, $\ell_i = z_i - \Delta_i/2$ and $u_i = z_i + \Delta_i/2$, for $0 \leq i < K$.

Range searching in any variant of K-d trees is straightforward. When visiting a node x that discriminates w.r.t. the j-th coordinate, we must compare x_j with the j-th range $[\ell_j, u_j]$ of the query. If the query range is totally above (or below) that value, we must search only the right subtree (respectively, left) of that node. If, on the contrary, $\ell_j \leq x_j \leq u_j$ then both subtrees must be searched; additionally, we must check whether x falls or not inside the query hyperrectangle. This procedure continues recursively until empty subtrees are reached.

We will measure the cost of range queries by the number of nodes of the random K-d tree visited during the search. If the number of points to be reported by the range search is P then the cost R_n of the range search will be of the form $\Theta(P + W_n)$, where W_n is the *overhead*.

3 Analysis of the cost of range searches

3.1 Bounding rectangles, slices and sliced partial match

The *bounding rectangle* $B(x) = [l_0(x), u_0(x)] \times \ldots \times [l_{K-1}(x), u_{K-1}(x)]$ of a point $x = (x_0, \ldots, x_{K-1})$ in a K-d tree t is the region of $[0,1]^K$ corresponding to the leaf replaced by x when x was inserted into t. Formally, it is defined as follows: (a) if x is the root of t then $B(x) = [0,1]^K$; (b) if $y = (y_0, \ldots, y_{K-1})$ is the father of x in t and y discriminates w.r.t. the j-th coordinate then: (b.1) if $x_j < y_j$ then $B(x) = [l_0(y), u_0(y)] \times$

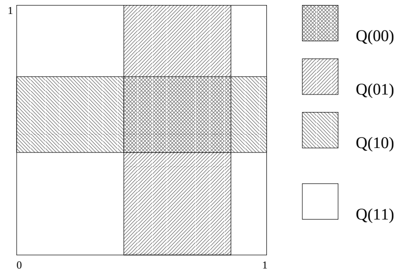

Figure 1: Example of the proper slices induced by a query Q

$\ldots \times [l_j(y), y_j] \times \ldots \times [l_{K-1}(y), u_{K-1}(y)]$, and (b.2) if $x_j \geq y_j$ then $B(x) = [l_0(y), u_0(y)] \times \ldots \times [y_j, u_j(y)] \times \ldots \times [l_{K-1}(y), u_{K-1}(y)]$.

Lemma 1. *A point x with bounding rectangle $B(x)$ is visited by a range search with query hyperrectangle Q if and only if $B(x)$ intersects Q.*

Proof. See [5, 7]. □

In order to relate the performance of range searches with the performance of partial matches, we need to introduce several notions, beginning with that of *slice*. Given a bitstring $w = (w_0, \ldots, w_{K-1})$ of length K, the slice Q_w is the K-dimensional hyperrectangle defined by

$$Q_w = [\ell'_0, u'_0] \times [\ell'_1, u'_1] \times \cdots \times [\ell'_{K-1}, u'_{K-1}],$$

where $[\ell'_i, u'_i] = [\ell_i, u_i]$ if $w_i = 0$ and $[\ell'_i, u'_i] = [0, 1]$ if $w_i = 1$. Notice that $Q_{00\ldots0} = Q$ and $Q_{11\ldots1} = [0, 1]^K$.

Another useful notion is that of *proper slice*. The proper slice $\hat{Q}_w$ is the hyperregion defined by

$$\hat{Q}_w = Q_w - \bigcup_{v<w} Q_v,$$

that is, the region that results when all the slices properly contained within Q_w are subtracted from it (see Figure 3.1; $\hat{Q}_w$ is denoted $Q(w)$ in this and other figures). The only proper slice consisting of a simple connected region is $\hat{Q}_{00\ldots0} = Q_{00\ldots0} = Q$; in general, $\hat{Q}_w$ consists of $2^{\mathrm{order(w)}}$ connected subregions, where $\mathrm{order}(w)$ is the number of 1s in the bitstring w.

The most important concept in this subsection is that of *sliced partial match*. Given a query hyperrectangle Q, a bitstring w and a point $y \in [0, 1]^K$, a sliced partial match acts as a standard partial match with query $q = (q_0, q_1, \ldots, q_{K-1})$ where $q_i = y_i$ if $w_i = 1$ and $q_i = *$ if $w_i = 0$ (hence the specification pattern of the partial match is w), but contrary to a standard partial match we will only count ("report") the visited points x in the data structure such that $x \in \hat{Q}_w$.

To every (sliced) partial match with point y and specification pattern w we associate the hyperplane $H(y, w)$ defined by

$$H(y,w) = \{x \in [0,1]^K \mid \forall i : w_i = 1 \implies x_i = y_i\}.$$

For instance, if $K = 2$ then $H(y, 00) = [0,1]^K$, $H(y, 11) = \{y\}$, $H(y, 01)$ is the line passing through y parallel to the horizontal axis and $H(y, 10)$ is the line passing through y parallel to the vertical axis.

Lemma 2. *A point x with bounding rectangle $B(x)$ is visited and reported by a sliced partial match with query hyperrectangle Q, specification w, and point y if and only if $x \in \hat{Q}_w$ and the bounding rectangle $B(x)$ intersects the hyperplane $H(y, w)$.*

Proof. The proof is immediate from Lemma 1. Notice that a sliced partial match behaves as a range query in which the hyperrectangle query "degenerates" to the hyperplane $H(y, w)$, but reports only those points that belong to $\hat{Q}_w$. □

3.2 The combinatorial characterizations

In this subsection we state several relations between the cost $R(t)$ of an orthogonal range search in a K-d tree t and the performance $P_w(t, y)$ of a sliced partial match with specification pattern w and query point y in a K-d tree t. The implicit query hyperrectangle Q is the same for both the range search and the sliced partial match.

Theorem 3. *Given a query Q with corners $v_0, v_1, \ldots, v_{2^K-1}$ and a K-d tree t,*

$$R(t) \leq \sum_{0 \leq j < 2^K} \sum_{w \in (0+1)^K} P_w(t, v_j).$$

Proof. Consider a point x visited by a range search with query Q. Let w be the index of the proper slice that contains x, i.e., $x \in \hat{Q}_w$. Recall that since x is visited by the range search we have $B(x) \cap Q \neq \emptyset$ (Lemma 1). Therefore, by Lemma 2, it suffices to show that if $B(x)$ intersects Q then there exists at least one corner v_j of Q such that the hyperplane $H(v_j, w)$ does intersect $B(x)$.

If $B(x)$ contains any of the corners of Q then the statement above is clearly true: since the hyperplane $H(v_j, w)$ contains v_j, it must intersect $B(x)$. If $B(x)$ does not contain any corner of Q, there are two possibilities to consider: either $B(x)$ is entirely within Q, or $B(x)$ intersects one or

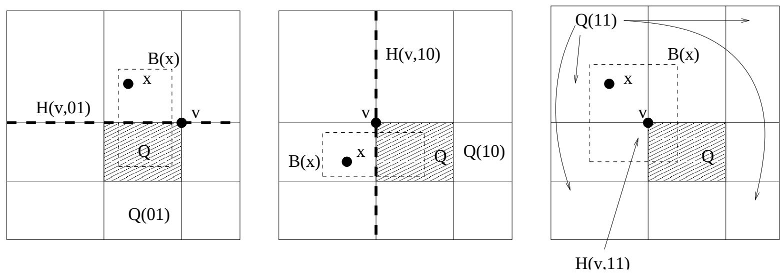

Figure 2: Graphical illustration of the proof of Theorem 3

more faces of Q. If $B(x)$ is totally inside Q then $w = 0 \ldots 0$ and indeed $H(v_j, 0 \ldots 0) = [0,1]^K$ intersects $B(x)$ for any corner v_j. On the other hand, if $B(x)$ intersects one or more faces of Q but does not contain a corner nor it is contained inside Q then $w \neq 11 \ldots 1$ and $\hat{Q}_w$ must "contact" one of the intersected faces, in the sense that the face is a boundary of $\hat{Q}_w$. Let f be such face. Now, the hyperplane $H(v_j, w)$ contains this face (and hence it intersects $B(x)$), provided that v_j is any corner of the face f. $\square$

Theorem 4. *Given a query Q with center at z and a K-d tree t,*

$$R(t) \geq \sum_{w \in (0+1)^K} P_w(t, z).$$

Proof. The statement of the theorem is immediate, once we show that whenever a point x is reported by a sliced partial match with parameters w and z then it is also visited by the range search with query Q. Formally, we have to show that if $x \in \hat{Q}_w$ and $H(z, w)$ intersects the bounding rectangle $B(x)$ of x then $B(x)$ also intersects Q. Indeed, if $w = 00 \ldots 0$ then the statement trivially holds since $x \in Q$. On the other hand, if $w \neq 00 \ldots 0$ then $H(z, w)$ and $\hat{Q}_w$ are disjoint. Since $B(x)$ intersects $\hat{Q}_w$ (x is part of both by hypothesis) the only way for $B(x)$ to intersect $H(z, w)$ is to intersect Q too. $\square$

Theorem 5. *Given a query Q with center at z divide $[0,1]^K$ into 2^K quadrants $C_0, C_1, \ldots, C_{2^K-1}$. Let $R^{(i)}(t)$ denote the number of points of the i–th quadrant visited by a range search with query Q in the K-d tree t. Similarly, let $P_w^{(i)}(t, y)$ denote the number of points of the i–th quadrant reported by sliced partial match with pattern w and point y in the K-d tree t. Let v_i be the unique corner of Q belonging to the i–th quadrant. Then*

$$R^{(i)}(t) = \sum_{w \in (0+1)^*} P_w^{(i)}(t, v_i).$$

Sketch of proof. For a point x belonging to the i-th quadrant and the proper slice $\hat{Q}_w$, the intersection of $H(v_i, w)$ with $B(x)$ implies the intersection of Q with $B(x)$. On the other hand, if $B(x)$ intersects Q there is at least one corner v such that $H(v, w)$ intersects $B(x)$; it is not difficult to see that one of these corners must be v_i, the unique corner of Q in the i-th quadrant. $\square$

3.3 Boundary effects: a digression

Up to now, we have not considered what happens if query hyperrectangles partially fall off the boundaries of $[0,1]^K$. If the the center z of the query is randomly drawn from $[0,1]^K$ this might happen. Also, the corners of the queries follow a shifted version of z's distribution (the amount of shift w.r.t. the i-th coordinate being $\Delta_i/2$ to the left or to the right depending of the corner). Furthermore, estimating the volumes of proper slices is more difficult. Alternatively, we might consider that query hyperrectangles never intersect the $[0,1]^K$ boundary by imposing that z is randomly drawn from the distribution that we obtain by mapping the continuous distribution on $[0,1]^K$ to $[\Delta_0/2, 1-\Delta_0/2] \times \cdots \times [\Delta_{K-1}/2, 1-\Delta_{K-1}/2]$. But again, the corners are distributed according to a shifted version of z's distribution. In any case, the corners of the queries are not necessarily distributed as the data points and the center of the query, and they cannot fall in some regions of $[0,1]^K$ (e.g., for $K=2$ if queries are allowed to fall partially off the boundaries the lower left corner falls in $[-\Delta_0/2, 1-\Delta_0/2] \times [-\Delta_1/2, 1-\Delta_1/2]$; if queries are not allowed to fall off the boundaries then the lower left corner can only fall in $[0, 1-\Delta_0] \times [0, 1-\Delta_1]$).

Thus the developments in the next subsection are only valid when the Δ's are small enough, since the perturbation due to the boundary and shifting effects is then negligible.

3.4 The expected cost of range search in relaxed K-d trees

The theorems of Subsection 3.2 show that the analysis of range search reduces to the analysis of sliced partial matches.

Theorem 6. *Let $\mathbb{E}[R_n]$ be the expected cost of a range search with a random query in a random K-d tree of size n. Let $\mathbb{E}[P_{n,w}]$ be the expected cost of a slice partial match with pattern w in a random K-d tree of size n, with respect to a random query and a random point. If both the data points and query centers are uniformly distributed in $[0,1]^K$ and the Δ's are small enough then*

$$\mathbb{E}[R_n] = \sum_{w \in (0+1)^*} \mathbb{E}[P_{n,w}].$$

Proof. Clearly $R(t) = \sum_{0 \le i < 2^K} R^{(i)}(t)$. Hence,

$$R(t) = \sum_{0 \le i < 2^K} \sum_{w \in (0+1)^*} P_w^{(i)}(t, v_i).$$

The next step is to take expectations on both sides of the equation above. Two points are worth noting: (1) since the center of the query z is uniformly distributed, each v_i is (almost) uniformly distributed (see Subsection 3.3); (2) the expected value $\mathbb{E}\left[P_{n,w}^{(i)}\right] = \mathrm{Vol}(C_i) \cdot \mathbb{E}\left[P_{n,w}\right]$, where $\mathrm{Vol}(C_i)$ is the probability that, given a randomly drawn point z, a second random point falls in the i-th quadrant defined by z. Since $\sum_{0 \le i < 2^K} \mathrm{Vol}(C_i) = 1$ the result follows. □

The problem is now to analyze the expected cost of sliced partial matches in random(ized) relaxed K-d trees. It easily follows from the analysis of the expected cost of standard partial matches in random relaxed K-d trees [12]. Our next theorem gives the expected cost of sliced partial matches in random relaxed K-d trees.

Theorem 7. *If $w \neq 00\ldots0$ and $w \neq 11\ldots1$, the expected cost $\mathbb{E}\left[P_{n,w}\right]$ of a sliced partial match in a random relaxed K-d tree of size n w.r.t. a random point and the pattern w is*

$$\mathbb{E}\left[P_{n,w}\right] = \mathit{Vol}(\hat{Q}_w) \cdot \beta(\rho) \cdot n^{\alpha(\rho)} + \mathcal{O}(1),$$

where $\rho = \mathit{order}(w)/K$, $\alpha \equiv \alpha(x) = (\sqrt{9-8x} - 1)/2$, $\beta(x) = \Gamma(2\alpha + 1)/((1-x)(\alpha+1)\alpha^3\Gamma^3(\alpha))$, and $\mathit{Vol}(\hat{Q}_w)$ is the probability that a point falls inside the proper slice $\hat{Q}_w$ of a randomly centered query. Furthermore, $\mathbb{E}\left[P_{n,00\ldots0}\right] = \mathit{Vol}(\hat{Q}_{00\ldots0}) \cdot n$ and $\mathbb{E}\left[P_{n,11\ldots1}\right] = 2 \cdot \mathit{Vol}(\hat{Q}_{11\ldots1}) \cdot (H_{n+1} - 1)$, where $H_n = \sum_{1 \le j \le n} 1/j = \log n + \gamma + \mathcal{O}(1/n)$ denotes the n-th harmonic number.

Sketch of proof. It is relatively easy to establish that $\mathbb{E}\left[P_{n,w}\right] = \mathrm{Vol}(\hat{Q}_w) \cdot \mathbb{E}\left[P_{n,w}^{[\mathrm{std}]}\right]$, where $\mathbb{E}\left[P_{n,w}^{[\mathrm{std}]}\right]$ is the expected number of visited nodes in a standard partial match with query pattern w on a random K-d tree of size n. The statement of the theorem follows from the known asymptotic estimates of $\mathbb{E}\left[P_{n,w}^{[\mathrm{std}]}\right]$ (see [12]). □

Corollary 8. *The expected cost of a random centered range search of sides $\Delta_0, \ldots, \Delta_{K-1}$ in a random relaxed K-d tree of size n satisfies, for uniformly and independently drawn data points and centers of the queries (with queries not falling off the boundaries), and if the values of the Δ's are small enough,*

$$\mathbb{E}\left[R_n\right] = \Delta_0 \cdots \Delta_{K-1} \cdot n + \sum_{1 \le j < K} c_j \cdot n^{\alpha(j/K)} + 2 \cdot (1-\Delta_0) \cdots (1-\Delta_{K-1}) \cdot \log n + \mathcal{O}(1),$$

where

$$c_j = \beta(j/K) \cdot \sum_{w:order(w)=j} \left(\prod_{i:w_i=0} \Delta_i \right) \cdot \left(\prod_{i:w_i=1} (1 - \Delta_i) \right).$$

Proof. The corollary easily follows from Theorem 6 and the asymptotic estimates given in Theorem 7. Finally, observe that for the uniform distribution we have

$$\mathrm{Vol}(\hat{Q}_w) = \left(\prod_{i:w_i=0} \Delta_i \right) \cdot \left(\prod_{i:w_i=1} (1 - \Delta_i) \right).$$

since we do not allow the query hyperrectangles to intersect the $[0,1]^K$ boundaries. ☐

Notice that the term $\Delta_0 \cdots \Delta_{K-1} \cdot n$ in $\mathbb{E}[R_n]$ is the expected number of reported points and hence the overhead is $\mathcal{O}(n^{\alpha(1/K)})$.

4 Other multidimensional data structures

It is important to stress that no assumptions were made with respect to the way that discriminants are assigned during the construction of the tree, so all theorems of subsection 3.2 apply to standard K-d trees [1], squarish K-d trees [7], K-d-t trees [6] and other variants. It turns out that the theorems also apply to quadtrees [2] without change.

Furthermore, similar arguments to that of Theorem 7 are also valid, relating the expected cost of sliced partial matches to the expected cost of standard partial matches. In general, when $w \neq 00\ldots0$ and $w \neq 11\ldots1$ for the data structures mentioned above we have

$$\mathbb{E}[P_{n,w}] = \beta_w \cdot \mathrm{Vol}(\hat{Q}_w) \cdot n^{\alpha(\rho)} + \mathcal{O}(1),$$

where $\rho = \mathrm{order}(w)/K$, $\alpha(x) = 1 - x + \phi(x)$ and β_w is a constant depending on w. The expected cost of range search takes hence the form

$$\begin{aligned} \mathbb{E}[R_n] = \Delta_0 \cdots \Delta_{K-1} \cdot n + \sum_{1 \le j < K} c_j \cdot n^{\alpha(j/K)} \\ + 2 \cdot (1 - \Delta_0) \cdots (1 - \Delta_{K-1}) \cdot \log n + \mathcal{O}(1), \end{aligned} \tag{1}$$

where $c_j = \sum_{w:\mathrm{order}(w)=j} \beta_w \cdot \mathrm{Vol}(\hat{Q}_w)$.

Different data structures are characterized by different α's and β's. For standard K-d trees the necessary analysis is from [10] where it was shown that $\phi(x) < 0.07$ and that it is the unique real solution of

$$(\phi(x) + 3 - x)^x (\phi(x) + 2 - x)^{1-x} - 2 = 0.$$

No closed expression for the β's is given, but their values can be explicitly computed with some effort; for $K \leq 4$ all the numerical values are given in [10]. Squarish K-d trees [7] have optimal performance since $\phi(x) = 0$; however, the values of the β's are not yet known and the only way for the moment to compute them would be through experimental measurement. For quadtrees the analysis of partial match can be found in [9]; $\alpha(x)$ is the same as for K-d trees. But the β_w's depend only on K and the order of w. For $K = 2$, we have $\beta_{01} = \beta_{10} = \Gamma(2\alpha+2)/(2\alpha^3\Gamma^3(\alpha)) \approx 1.5950991$, but no explicit form is given for higher dimensions.

Last, but not least, the theorems in Section 3 also apply to K-d tries, relaxed K-d tries and quadtries. These multidimensional data structures are space-oriented rather than data-oriented, since they induce partitions of the space that are independent of the data points (except that the recursive subdivision of the space stops when the corresponding region contains only one or no points). In order to apply the theorems of Section 3, we only need to assume that each internal node "contains" the middle point of the hyperplane(s) associated to the internal node, to meaningfully define slice partial matches in these data structures. The average cost of range searches in these digital data structures satisfies (1), but the β_w's involve a fluctuating periodic term (depending on n) of small amplitude and bounded by a constant. For instance, for relaxed K-d tries [12] we have $\alpha = \alpha(x) = \log_2(2-x)$ and $\beta_w = \beta(\mathrm{order}(w)/K)$ with

$$\beta(x) = \frac{1}{\log 2}\Big((\alpha-1)\Gamma(-\alpha) + \delta(\log_2 n)\Big),$$

and $\delta(\cdot)$ a periodic function of period 1, mean 0 and small amplitude that also depends on α.

References

[1] J.L. Bentley. Multidimensional binary search trees used for associative retrieval. *Communications of the ACM*, 18(9):509–517, 1975.

[2] J.L. Bentley and R.A. Finkel. Quad trees: A data structure for retrieval on composites keys. *Acta Informatica*, 4:1–9, 1974.

[3] J.L. Bentley and J.H. Friedman. Data structures for range searching. *ACM Computing Surveys*, 11(4):397–409, 1979.

[4] J.L. Bentley and D.F. Stanat. Analysis of range searches in quad trees. *Information Processing Letters*, 3(6):170–173, 1975.

[5] P. Chanzy, L. Devroye, and C. Zamora-Cura. Analysis of range search for random k-d trees. *Acta Informatica*, 37:355–383, 2001.

[6] W. Cunto, G. Lau, and Ph. Flajolet. Analysis of kdt-trees: kd-trees improved by local reorganisations. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Workshop on Algorithms and Data Structures (WADS'89)*, volume 382 of *Lecture Notes in Comput. Sci.*, pages 24–38. Springer-Verlag, 1989.

[7] L. Devroye, J. Jabbour, and C. Zamora-Cura. Squarish k-d trees. *SIAM Journal on Computing*, 30:1678–1700, 2000.

[8] A. Duch, V. Estivill-Castro, and C. Martínez. Randomized k-dimensional binary search trees. In K.-Y. Chwa and O. H. Ibarra, editors, *Int. Symposium on Algorithms and Computation (ISAAC'98)*, volume 1533 of *Lecture Notes in Comput. Sci.*, pages 199–208. Springer, 1998.

[9] Ph. Flajolet, G. Gonnet, C. Puech, and J. M. Robson. Analytic variations on quadtrees. *Algorithmica*, 10:473–500, 1993.

[10] Ph. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[11] H.M. Mahmoud. *Evolution of Random Search Trees.* J. Wiley & Sons, 1992.

[12] C. Martínez, A. Panholzer, and H. Prodinger. Partial match queries in relaxed multidimensional search trees. *Algorithmica*, 29(1–2):181–204, 2001.

[13] R. L. Rivest. Partial-match retrieval algorithms. *SIAM Journal on Computing*, 5(1):19–50, 1976.

[14] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, 1990.

[15] Y.V. Silva-Filho. Average case analysis of region search in balanced k-d trees. *Information Processing Letters*, 8(5):219–223, 1979.

A Experiments

We have conducted a series of preliminary experiments to validate the theoretical analysis of the previous sections and to explore its limitations, when the input data does not fulfill some of the hypotheses of the random model.

In each experiment, each "sample point" consisted of a random relaxed and standard K-d tree of size n and dimension K, built from the same sequence of random insertions. Up to Q range searches with random queries of given edge lengths were requested in both trees. And this was repeated for T pairs of trees of each size and dimension.

The legends of the plots indicate the different parameter values: dimension (K), size (n), number of trees per size (T), number of queries per tree (Q) and edge lengths (Δ). Each plot depicts the empirical expected cost of range search against the theoretical predicted value.

The experiments suggest that the variance of the cost of range search is high. We conjecture that if $\mathbb{E}[R_n] = a \cdot n + b \cdot n^{\alpha} + o(n^{\alpha})$, the variance of R_n is $\Theta(n^{2\alpha})$.

We have also performed experiments using the same setting as for the first set of experiments, but the data points and query centers were drawn from a clustered distribution (see for example Figure 9). In order to generate these clustered distributions, a number of "clusters" ($c = 10$ in the experiments) is fixed in advance and the centers of the clusters uniformly and independently generated. For each sampled point, a cluster is selected at random with identical probability and each coordinate of the point is then generated according to a normal law whose mean is the corresponding coordinate of the center of the cluster. The standard deviations σ_i of the normal laws were taken all identical to 0.05. Data points falling outside the boundaries were not allowed. The centers of the queries were generated according to the same distribution.

There are two problems with Corollary 8 and similar results for non-uniform distributions and non-independent data. First, it is very difficult, if not impossible, to compute precisely the volumes of the proper slices. We have prepared a program to compute the approximate values of these volumes for given Δ's, by repeatedly "throwing" queries with the given edge lengths and a large number of data points for each query, keeping frequency counts of the event "point falls in $\hat{Q}_w$". For instance, when $K = 2$ and $\Delta = [0.01, 0.01]$ we have the estimates $\mathrm{Vol}(\hat{Q}_{00}) = 0.000868$, $\mathrm{Vol}(\hat{Q}_{01}) = 0.022335$, $\mathrm{Vol}(\hat{Q}_{10}) = 0.019432$ and $\mathrm{Vol}(\hat{Q}_{11}) = 0.957365$, which are significantly different from the volumes for the given values of Δ in the uniform distribution.

The second problem is that Corollary 8 is not true for a non-uniform distribution, because the combined "perturbations" due to boundary effects and the shifting and correlation in the distribution of the corners (see Sub-

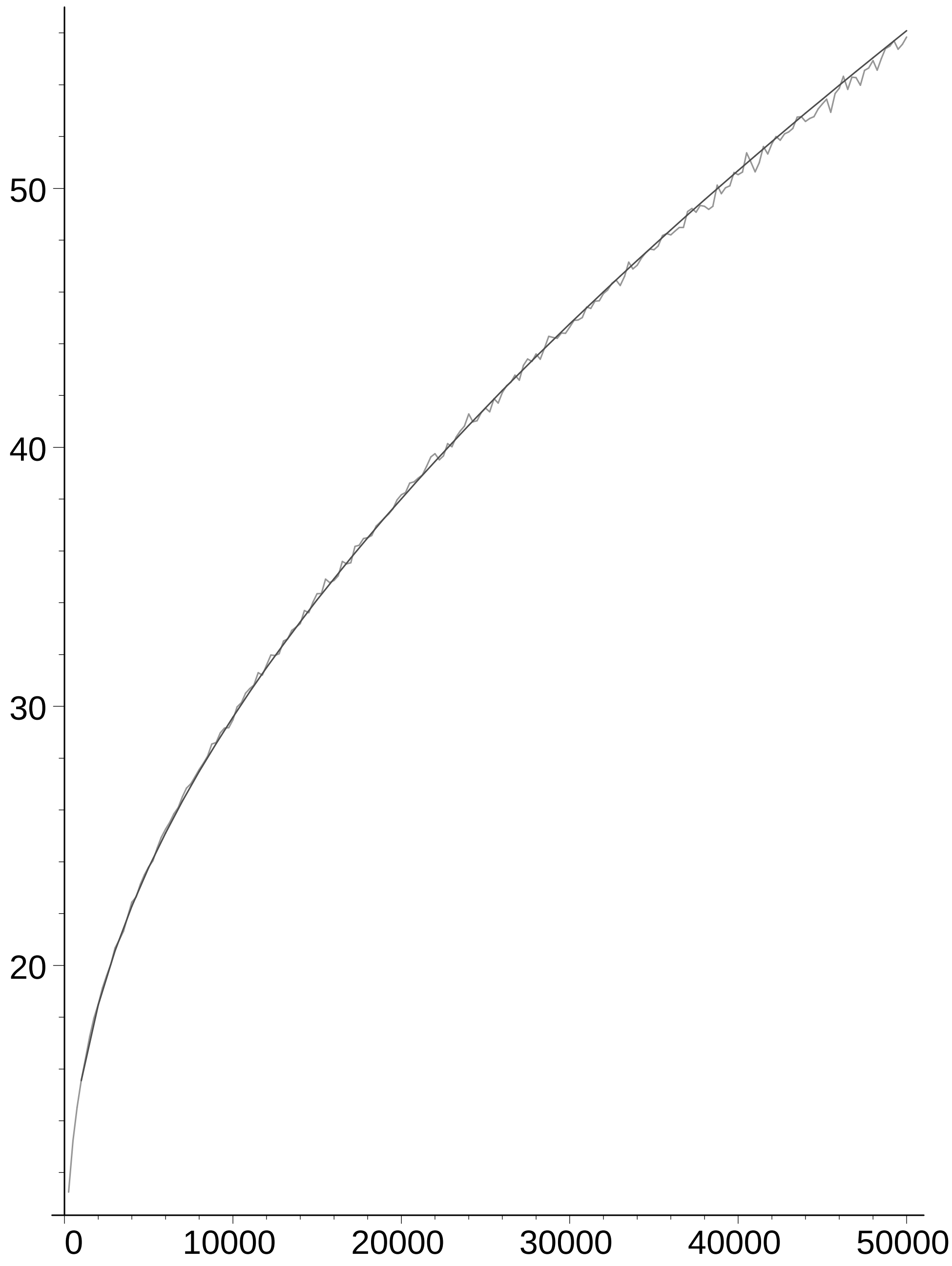

Figure 3: Relaxed K-d trees ($K = 2$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01]$)

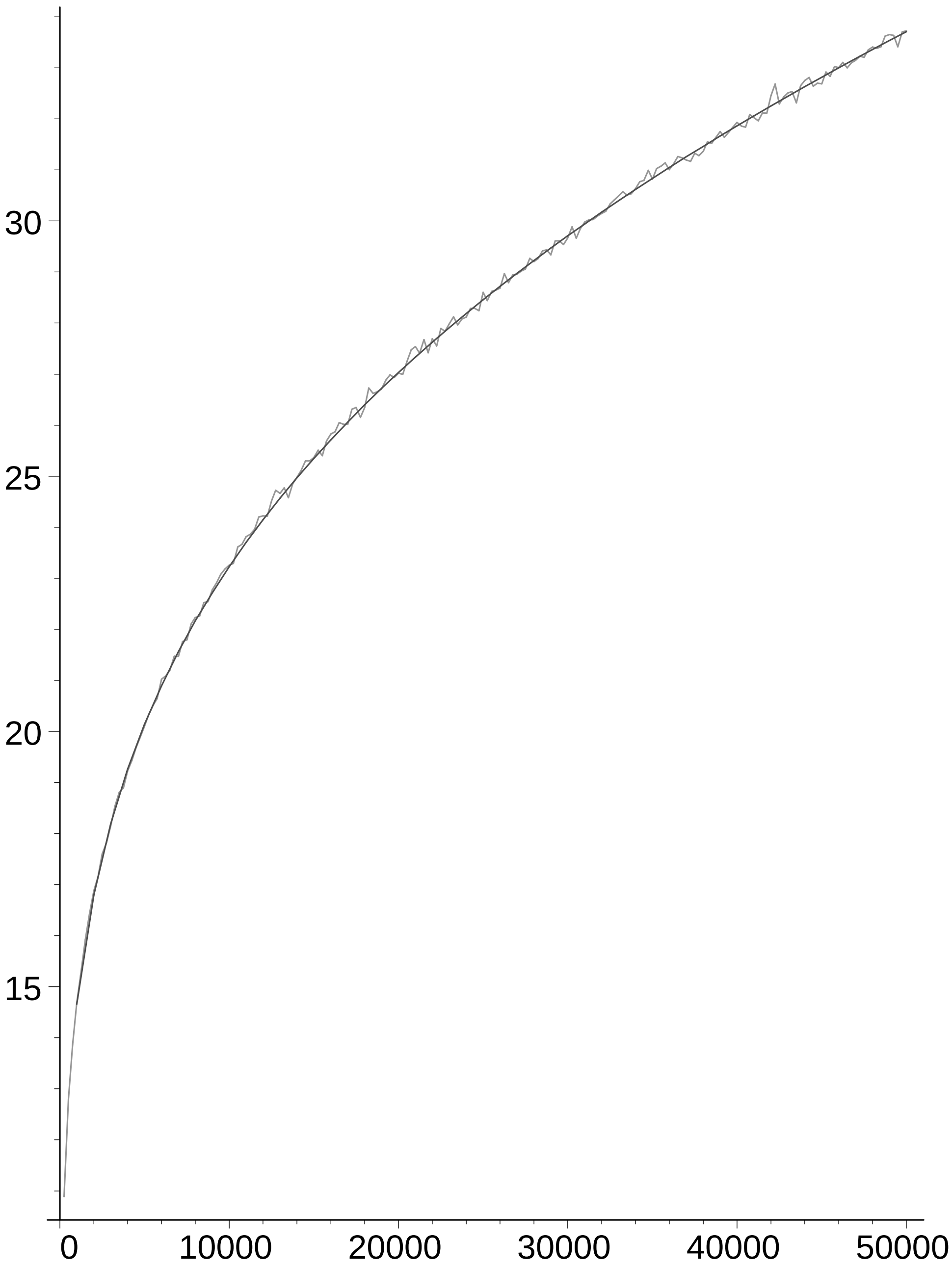

Figure 4: Relaxed K-d trees ($K = 3$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01, 0.01]$)

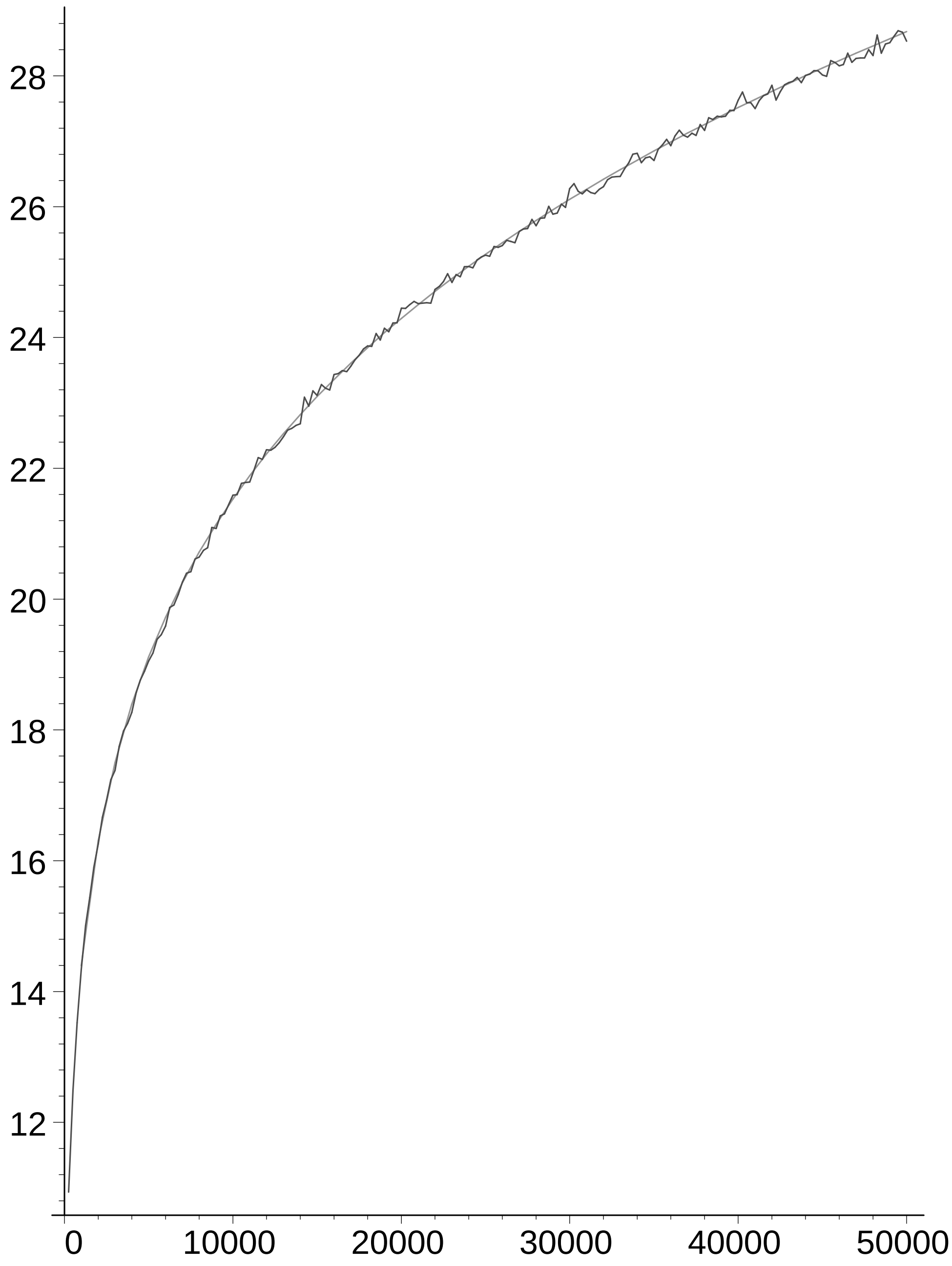

Figure 5: Relaxed K-d trees ($K = 4$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01, 0.01, 0.01]$)

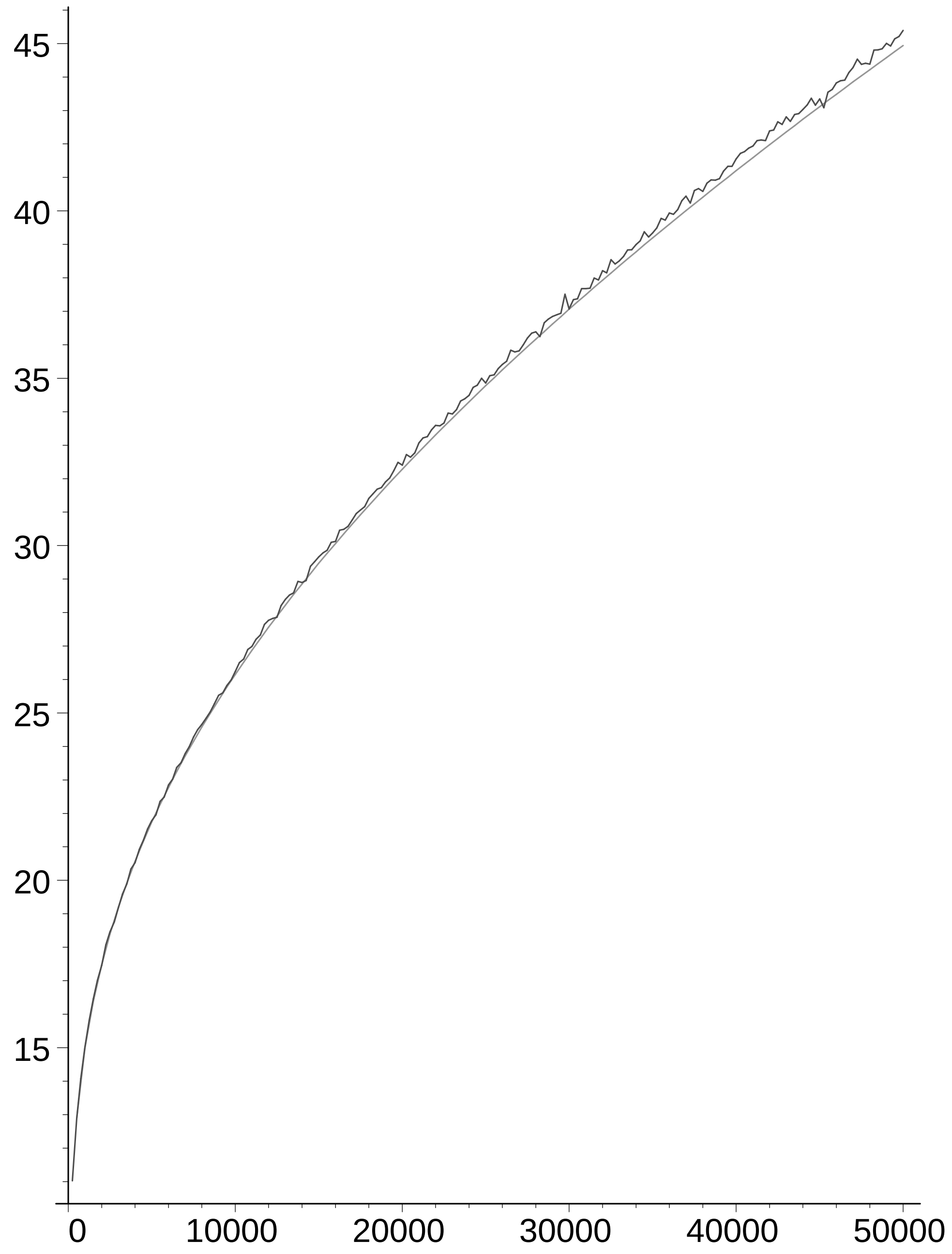

Figure 6: Standard K-d trees ($K = 2$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01]$).

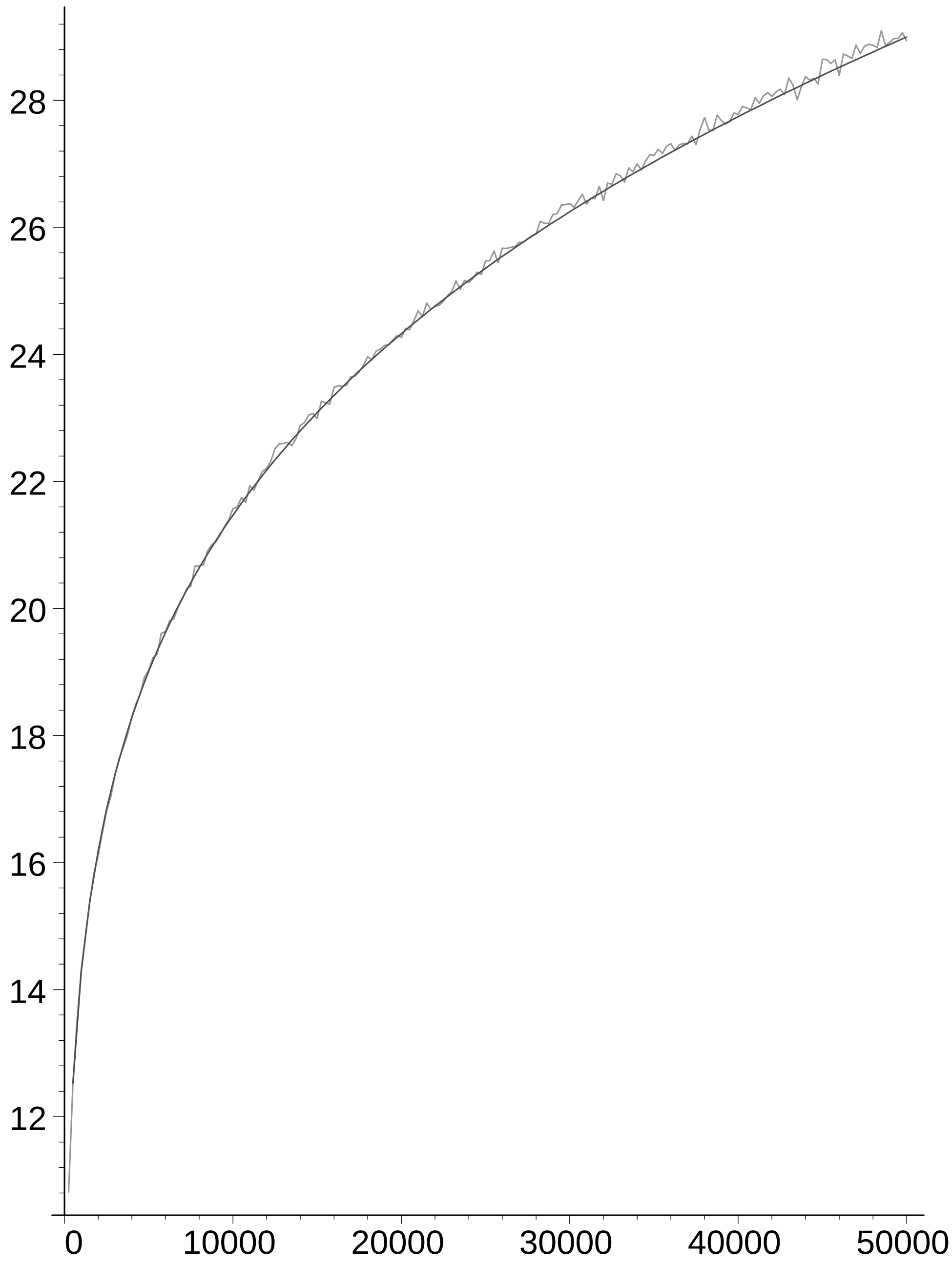

Figure 7: Standard K-d trees ($K = 3$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01, 0.01]$).

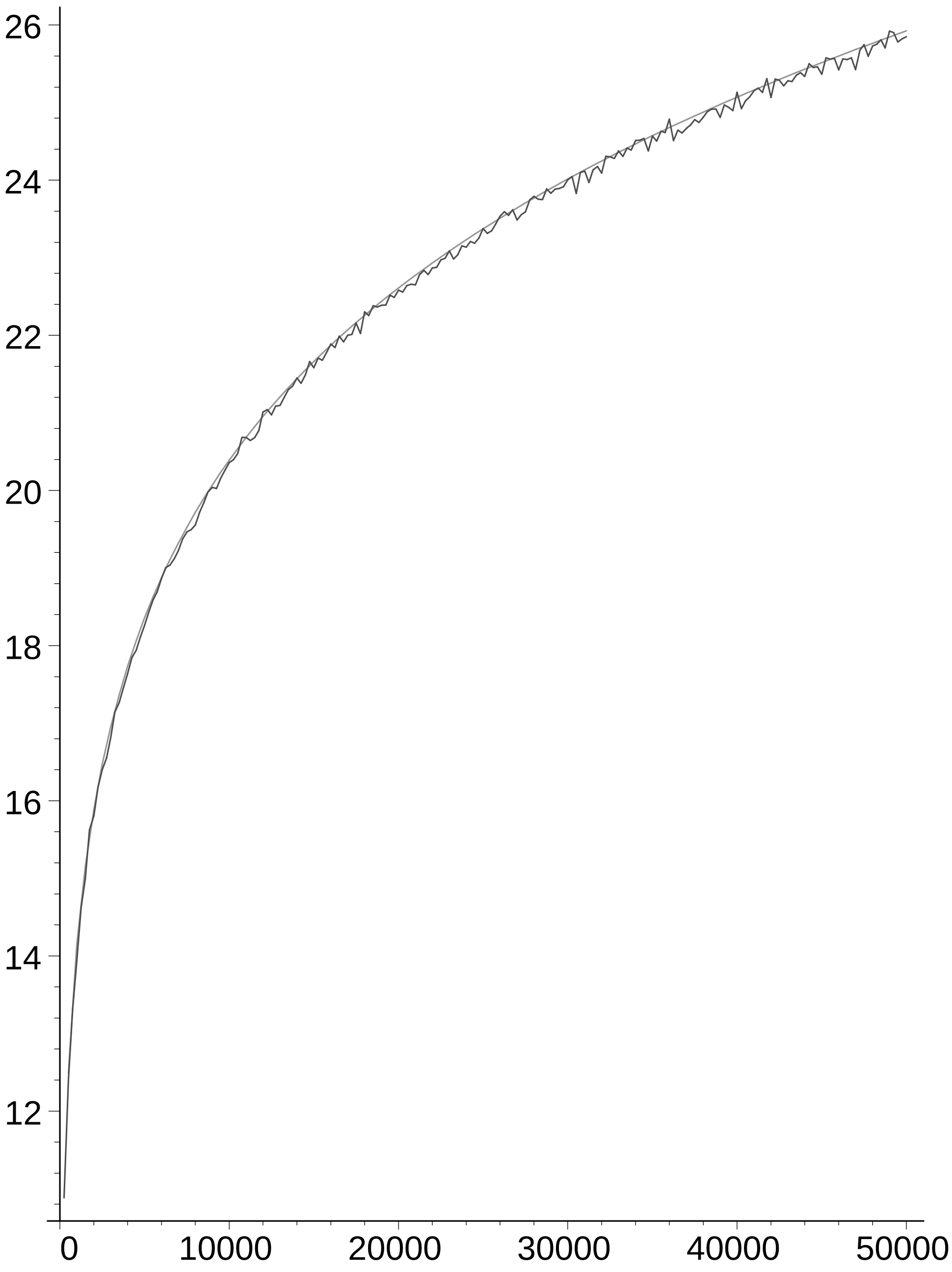

Figure 8: Standard K-d trees ($K = 4$, $n \leq 50000$, $T = 300$, $Q = 100$, $\Delta = [0.01, 0.01, 0.01, 0.01]$).

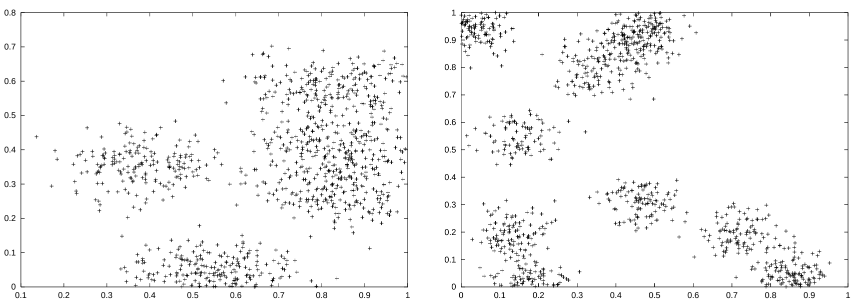

Figure 9: Two clustered distributions of $n = 1000$ points in $[0,1]^2$ (left: $c = 5, \sigma = [0.1, 0.05]$; right: $c = 10, \sigma = [0.05, 0.05]$).

section 3.3).

The plots for clustered input show the results of the experiments together with: a) the estimates given by Corollary 8 as if the points where uniformly distributed; b) the estimates obtained by plugging the “experimental” values of the proper slices’ volumes (see above) into the formula of Corollary 8.

The simulation programs where written in C (using the GNU `gcc-2.8.1` compiler), and run under Solaris 5.7 in a Sun Ultra 5 workstation. AWK scripts where used as a front-end to the C program for easier interaction and for data analysis. The plots were produced with Maple 6 and `gnuplot`.

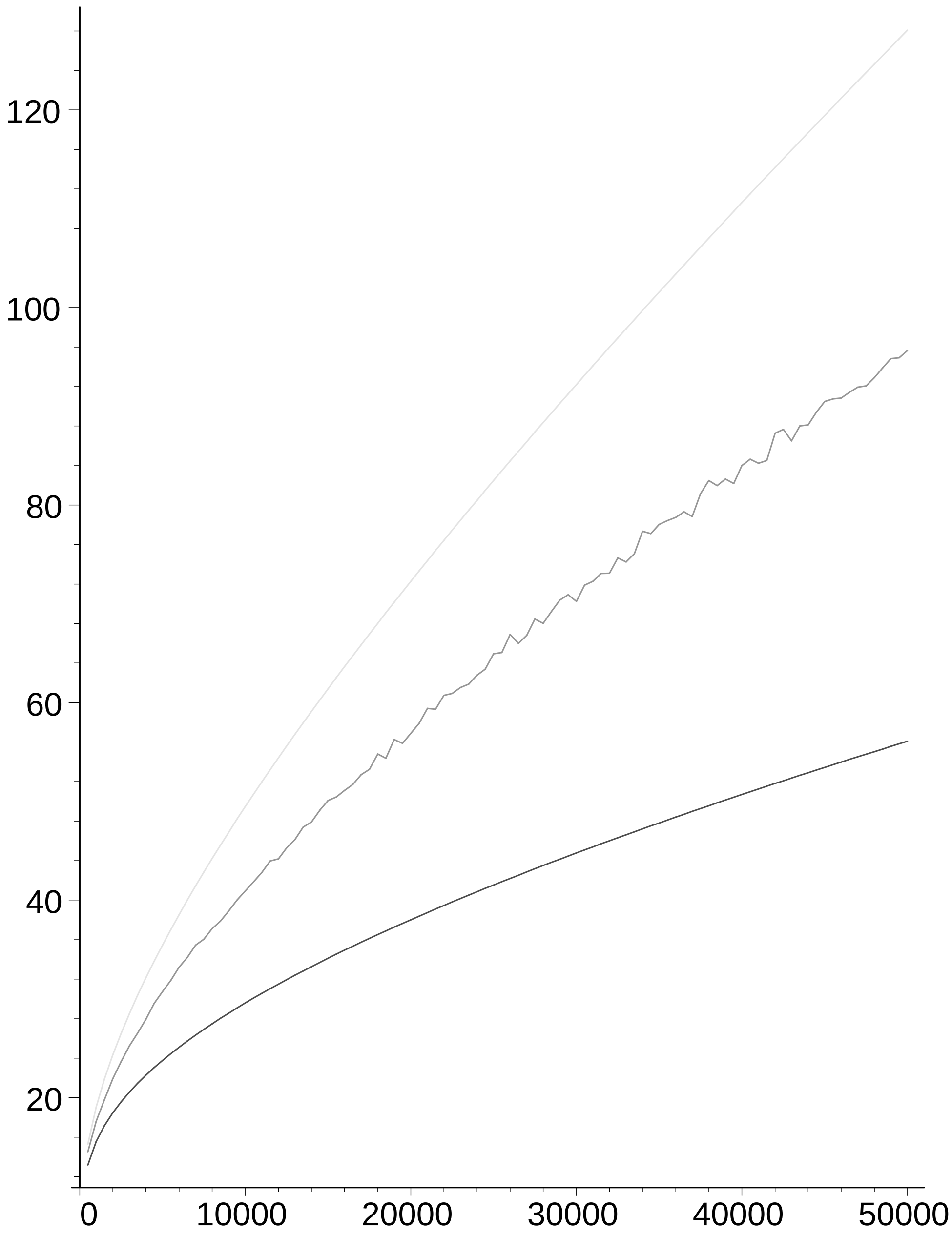

Figure 10: Relaxed K-d trees (clustered distribution, $K = 2$, $n \leq 50000$, $T = 50$, $Q = 1000$, $\Delta = [0.01, 0.01]$). Curve above: theoretical prediction with estimated proper slices' volumes; middle: empirically measured cost; below: theoretical prediction with uniform estimates for proper slices' volumes.

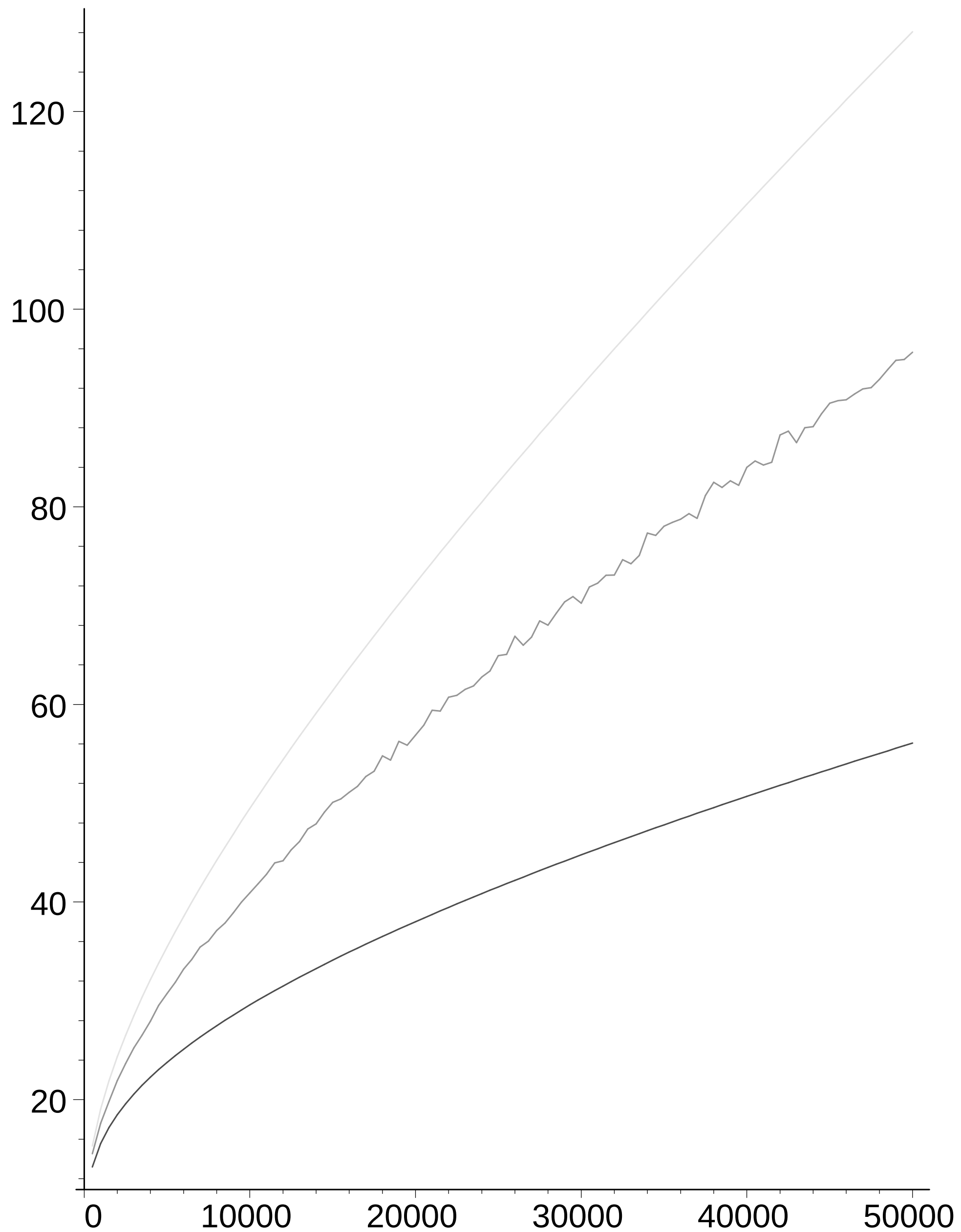

Figure 11: Standard K-d trees (clustered distribution, $K = 2$, $n \leq 50000$, $T = 50$, $Q = 1000$, $\Delta = [0.01, 0.01]$). Curves are as in previous figure (Figure 10).