On Dominance Reporting in 3D

Peyman Afshani
School of Computer Science
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada
pafshani@uwaterloo.ca

Abstract

In this paper, we study the 3D dominance reporting problem in different models of computations and offer optimal results in the pointer machine and the external memory models and a near optimal result in the RAM model; all our results consume linear space. We can answer queries in $O(\log n + k)$ time on a pointer machine, with $O(\log_B n + k/B)$ I/Os in the external memory model and in $O((\log\log n)^2 + \log\log U + k)$ time in the RAM model and in a $U \times U \times U$ integer grid. These improve the results of various papers, such as Makris and Tsakalidis (IPL'98), Vengroff and Vitter (STOC'96) and Nekrich (SOCG'07). Here, n, k and B are the input, output and block size respectively. With a $\log^3 n$ fold increase in the space complexity these can be turned into orthogonal range reporting algorithms with matching query times, improving the previous orthogonal range searching results in the external memory and RAM models. Using our 3D results as base cases, we can provide improved orthogonal range reporting algorithms in $\mathbb{R}^d$, $d \geq 4$. We use randomization only in the preprocessing part and our query bounds are all worst case.

1 Introduction

Let P be a set of n points in $\mathbb{R}^d$. In the dominance reporting problem we are given a query point $q = (q_1, \ldots, q_d)$ and we are asked to find all the points $p = (x_1, \ldots, x_d) \in P$ such that $x_i < q_i$, $1 \leq i \leq d$. Dominance reporting is one of the important problems in the orthogonal range searching area; it emerges naturally when studying various problems regarding orthogonal objects [1], it is an important special case of orthogonal range searching (which has been studied extensively, see [2, 3, 4, 5]) and many times it is used as a basis of various orthogonal range searching algorithms [6, 7, 8].

Previous results. Previously, in the pointer machine model, we were quite close to the optimal answer. Makris and Tsakalidis [9] had shown it is possible to achieve the query time of $O(\log n \log\log n + k)$ with linear space, improving an old result from 1987 [10] (here, n and k are the input size and output size respectively). Also in the same paper [9], they achieve the query time of $O((\log\log U)^2 \log\log\log U + k \log\log U)$ in the RAM model and with linear space (for points in a $U \times U \times U$ integer grid). Further results in the RAM model include an algorithm with linear space and query time of $O(\frac{\log n}{\log\log n} + k)$ [11] assuming integer inputs. However, these are not the fastest data structures, since if we allow $O(n \log n)$ space we can achieve the query time of $O((\log\log n)^2 + \log\log U + k)$ [6]. In the external memory model, there are much fewer results and it is believed that solving such orthogonal range reporting problems in this model is more difficult than the main memory model [7]. Currently, the best algorithm in the traditional external memory model uses $O(n \log n)$ space and can answer queries with optimal $O(\log_B n + k/B)$ I/Os [8, 12] (here B is the block size). Very recently, Nekrich has proposed another algorithm using $O(n \log_B n)$ space which can answer rectangular point location queries in a $U \times U$ grid with $O(\log\log_B U + (\log\log n)^2)$ I/Os [13]. Using this, he provides another dominance reporting algorithm with $O(n \log n)$ space and $O(\log\log_B U + (\log\log n)^2 + k/B)$ query I/Os.

Model	Space	Query complexity	Source
pointer machine	$O(n)$	$O(\log n \log\log n + k)$	[9]
pointer machine	$O(n \log n)$	$O(\log n + k)^*$	[10]
pointer machine	$O(n)$	$O(\log n + k)^*$	this paper
RAM	$O(n)$	$O((\log\log U)^2 \log\log\log U + k \log\log U)$	[9]
RAM	$O(n)$	$O(\frac{\log n}{\log\log n} + k)$	[11]
RAM	$O(n \log n)$	$O((\log\log n)^2 + \log\log U + k)$	[6]
RAM	$O(n)$	$O((\log\log n)^2 + \log\log U + k)$	this paper
external memory	$O(n \log n)$	$O(\log_B n + k/B)^*$	[8]
external memory	$O(n)$	$O(\log_B n + k/B)^*$	this paper
EM & integer grid	$O(n \log n)$	$O(\log\log_B U + (\log\log n)^2 + k/B)$	[13]
EM & integer grid	$O(n \log_B n)$	$O(\log\log_B U + (\log\log n)^2 + k/B)$	this paper

Table 1: Results on 3D dominance reporting problem. Here, n, k, B are input, output and block size respectively. The results in RAM assume the input is from a $U \times U \times U$ integer grid. Optimal query complexities are marked with a star. The last two rows assume an external memory model of computation in the RAM model where the input is on a $U \times U \times U$ integer grid.

Model	Dimension	Space	Query complexity	Source
RAM	$d = 3$	$O(n \log^4 n)$	$O((\log\log n)^2 + \log\log U + k)$	[6]
RAM	$d = 3$	$O(n \log^3 n)$	$O((\log\log n)^2 + \log\log U + k)$	this paper
RAM	$d > 3$	$O(n \log^{d+1+\varepsilon} n)$	$O(\log^{d-3} n/(\log\log n)^{d-5} + k)$	[6]
RAM	$d > 3$	$O(n \log^{d-2+\varepsilon} n)$	$O(\log^{d-2} n/(\log\log n)^{d-2} + k)$	[4]
RAM	$d > 3$	$O(n \log^{d+\varepsilon} n)$	$O(\log^{d-3} n/(\log\log n)^{d-5} + k)$	this paper
external memory	$d = 3$	$O(n \log^4 n)$	$O(\log_B n + k/B)$	[8]
external memory	$d = 3$	$O(n \log^3 n)$	$O(\log_B n + k/B)$	this paper

Table 2: The fastest known orthogonal range reporting algorithms for $d \geq 3$. Here, n, k, B are input, output and block size respectively.

Our results. In this paper we solve the 3D dominance reporting problem in pointer machine and external memory models and match the fast $O((\log\log n)^2 + \log\log U + k)$ query time of Nekrich [6] with a linear space algorithm. In essence, our techniques allow us to reduce the dominance reporting problem to point location in planar rectilinear arrangements. For instance, this enables us to use the new point location data structure of Nekrich [13] as black box and obtain another algorithm using $O(n \log_B n)$ space and with $O(\log\log_B U + (\log\log n)^2 + k/B)$ query I/Os, assuming the input is a subset of $U \times U \times U$ integer grid.

A summary of our results in comparison with the previous work is shown in Table 1. Using our efficient dominance reporting algorithms as base cases we can obtain new algorithms for orthogonal range reporting in the RAM model. These improvements have been listed in Table 2

Unlike some previous algorithms, our methods use many common ideas and techniques with the other important cases of range searching, namely halfspace and simplicial range searching. In fact, an underlying implication of our techniques is that both halfspace range reporting and dominance reporting in 3D can be attacked within the same framework and using the same array of techniques and tools. We believe this is one of the important contributions of our paper since apparently this had eluded the previous researchers. For instance, we use the shallow cutting lemma provided by Agarwal et al. [14] for a general class of surfaces (which is inspired by Matoušek's shallow cutting lemma for halfspaces [15]) and observe that it leads to approximate levels of optimal size. A concept similar to approximate levels was previously employed by Vengroff and Vitter [8] and later by Nekrich [6] but only under the name of B-approximate boundaries and with non-optimal and complicated constructions.

Given this, one might wonder whether other fundamental theorems of halfspace or simplicial range searching such as the shallow partition theorem can also be proven in this context. We do not investigate these questions and for two reasons (aside from being outside the scope of this paper). First, as we noted, the latter challenge has already been undertaken by Agarwal et al. [14] and second, we show the existence of a novel partition type theorem (it is not a partition per se and only resembles the partition theorem in "spirit") for the dominance reporting problem; unfortunately (or fortunately) our proof technique neither is inspired nor resembles the original shallow partition theorem. Nonetheless, this theorem seems to be stronger than an analogous of the shallow partition theorem for the dominance reporting problem since it leads to an optimal external memory data structure (fortunately). A similar result for the halfspace range reporting problem has not been obtained yet [16].

As a consequence of our results, we can obtain two orthogonal range reporting algorithms consuming $O(n\log^3 n)$ space; one with $O((\log\log n)^2 + \log\log U + k)$ query time in the RAM model and another with $O(\log_B n + k/B)$ I/Os in the external memory model. Previously, best results consumed $O(n\log^4 n)$ space in both cases. We only use randomization in the preprocessing part and our query bounds are all worst case.

2 Preliminaries

For two points A and B in $\mathbb{R}^d$, we say A *dominates* B if and only if all the coordinates of A are greater than those of B. In this paper, we shall deal with a special form of geometric ranges that we call a *downward corner*, which is uniquely determined by a point $A \in \mathbb{R}^d$ (called *apex*) and contains all the points of $\mathbb{R}^d$ which are dominated by A. Let P be a set of n points in $\mathbb{R}^d$. To make the notation consistent, we will reserve $\mathtt{r}$ for downward corners and with an abuse of notation, sometimes we will use $\mathtt{r}$ to refer both to the geometric object and the subset of P inside the geometric object. We define an *approximate* k*-level* $\mathcal{L}_k$ as a set of downward corners with the following two properties: (i) any downward corner $\mathtt{r} \in \mathcal{L}_k$ must contain at most $c_1 k$ points of P and (ii) any downward corner $\mathtt{r}'$ which contains at most k points of P must be contained in a downward corner $\mathtt{r} \in \mathcal{L}_k$. Here, c_1 can be chosen to be an arbitrary constant (by our algorithms). The size of an approximate level is the number of its downward corners. For a set S of geometric objects, we will use the shorthand notation of S_x to refer to the subset of S intersecting another geometric object x. Finally, for the sake of simplicity of description, we assume the input point set and the query points are in general position; a restriction that can be overcome using standard tricks.

3 Optimal Approximate Levels

We define the level of a point $p \in \mathbb{R}^3$ to be the number of downward corners of S which contain p. As with the case of halfspaces, we define the $(\leq k)$-level to be the set of all the vertices the arrangement $\mathscr{A}$ formed by S with level at most k. Thus, the (≤ 0)-level of $\mathscr{A}$ contains all the vertices of the arrangement that are not inside any downward corner of S.

One crucial requirement of any optimal lemma on approximate levels is a linear bound on the size of the (≤ 0)-level of the arrangement formed by the set of geometric objects.

Lemma 1. *For a set S of n downward corners the size of the (≤ 0)-level is $O(n)$.*

Proof. Sweep a plane h parallel to the xy plane from $z = +\infty$ to $z = -\infty$. We will count the vertices of the (≤ 0)-level of the arrangement as they appear on this sweep plane.

The apex A of an element $\mathtt{r} \in S$ will appear on h when the z-coordinate of h becomes equal to the z-coordinate of A and it will disappear as soon as another point q' on h dominates A (in 2D sense). The crucial observation is that if a point disappears from h it no longer can contribute any new vertices to the (≤ 0)-level (Fig. 1(a,b,c)). So, at any moment we have an active set of downward corners on this plane with none dominating another; these points form a chain on this plane (Fig. 1(d)). Assume a new point c_{t+1} appears on this plane. If c_{t+1} creates j new vertices then it will have to dominate and remove $\Omega(j-4)$ active points from h (Fig. 1(c,e)). A simple charging argument implies that number of vertices of the (≤ 0)-level is $O(n)$. □ □

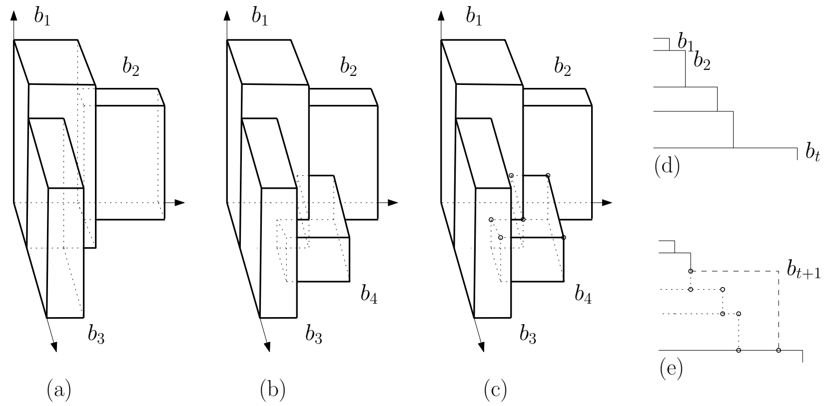


Figure 1: (a) Boxes b_1, b_2 and b_3 already swept (b) Sweep plane discovers b_4 and from this point b_1 can be ignored. (c) Marks denote the new vertices on the 0-level. (d,e) View on plane h

The shallow cutting lemma of Agarwal et al. [14] operates on a general class of surfaces and thus accepts a parameter $\phi(r)$ which is the worst case size of the (≤ 0)-level of any collection of r surfaces. The above lemma implies in our problem we have $\phi(r) = O(r)$. Combining this with the theorem of Agarwal et al. [14] we obtain the following lemma.

Lemma 2. *Given a set S of n downward corners in 3D and a parameter k, one can build a set $\mathcal{B}$ of $O\left(\frac{n}{k}\right)$ boxes which cover the $(\leq k)$-level of the arrangement formed by S where each box is intersected by $O(k)$ downward corners.*

Proof. With slight perturbations we can turn a downward corner into a continuous surface which fits the framework of Agarwal et al. [14] and use their shallow cutting lemma with $r = \frac{n}{k}$. The fact that the set $\mathcal{B}$ can be taken as a set of boxes follows from the vertical decomposition used by Agarwal et al. [14]. The source of randomness is the sampling technique used in the same paper. $\square$ $\square$

The above shallow cutting result can be used to construct approximate levels.

Lemma 3. *There exists approximate k-levels of size $O(\frac{n}{k})$, $1 \leq k \leq n$, for the dominance reporting problem.*

Proof. Let P be an input set of size n. For a point $p \in \mathbb{R}^3$, define an *upward corner with apex p* to be the subset of $\mathbb{R}^3$ which dominates p. Let S be the set of n upward corners determined by points of P as apexes and let $\mathscr{A}$ be the arrangement formed by S. A point reflection with origin can transform $\mathscr{A}$ into an arrangement of n downward corners, A', and thus we can use Lemma 2 and build a collection $\mathcal{B}'$ of $O(\frac{n}{k})$ boxes which cover the $(\leq k)$-level of $\mathscr{A}'$. Perform the point reflection on elements of $\mathcal{B}'$ and let $\mathcal{B}$ be the resulting set of boxes. For every box in $b \in \mathcal{B}$, place the vertex with the maximum coordinates, denoted with $m(b)$, in a set $\mathcal{C}$. We claim the set of downward corners defined by apexes in $\mathcal{C}$ is an approximate level.

Consider a downward corner $\mathtt{r}$ with apex A which contains less than k points of P. This means that there are less than k upward corners of S which contain A. The reflection A' of A by the origin lies in the $(\leq k)$-level of $\mathscr{A}'$ and thus there is a box $b \in \mathcal{B}$ which contains A. The downward corner defined by $m(b) \in \mathcal{C}$ contains $\mathtt{r}$.

On the other hand, Lemma 2 implies every box $b' \in \mathcal{B}'$ lies in the $(\leq O(k))$-level of $\mathscr{A}'$. Thus, every vertex of $b \in \mathcal{B}$ can dominate at most $O(k)$ vertices of P. $\square$ $\square$

Remarks. Agarwal et al. [14] do not discuss the construction time of their general shallow cutting theorem. Unfortunately, to achieve a decent bound on the construction time we will need to deal with the details of their construction which we feel would be distracting, despite the fact that for our set of ranges it is possible to simplify their argument. We postpone the details for the full version of the paper (simply claim a polynomial bound here) but it can be seen that we can employ the techniques used by

Ramos [17] and build the cutting in a gradual fashion (similar to the case for halfspaces) in $O(n \log n)$ expected time.

4 Solving the Dominance Reporting Problem

To solve the dominance reporting in the RAM and pointer machine we simply need a linear size data structure with polylogarithmic query time to combine it with our lemma on approximate levels. For instance, we can either use the data structure of Makris and Tsakalidis [9] or Eledsbrunner and Chazelle [10]. For the moment assume that we have access to a linear size data structure with $O(\log^2 n + k)$ query time.

Theorem 1. *Given a set of n points P in $\mathbb{R}^3$, dominance reporting queries can be answered in $O(\log n+k)$ worst case time in a pointer machine and using linear space.*

Proof. Let $\mathcal{A}$ be a data structure consuming linear space which can answer dominance reporting queries in $O(\log^2 n + k)$ time. Build an approximate $\log^2 n$-level $\mathcal{C}$. For every downward corner $\mathtt{r} \in \mathcal{C}$ implement the data structure $\mathcal{A}$ on the points contained in $\mathtt{r}$ and at the end build on extra copy for the whole point set P.

If $k = \Omega(\log^2 n)$ then we can use the data structure on P to answer the query in $O(\log^2 n + k) = O(k)$ time. Otherwise, one downward corner $\mathtt{r}$ in $\mathcal{C}$ will contain q. Finding q is known to be equivalent to a point location query in a 2D orthogonal arrangement [9, 6] and thus can be solved in $O(\log n)$ time. Since $\mathtt{r}$ will contain at most $\log^2 n$ points, the query can be answered in $O((\log \log^2 n)^2 + k) = O((\log \log n)^2 + k)$ time using the data structure implemented on $\mathtt{r}$. Combining all these results in a query time of $O(\log n + k)$. □ □

The exact same idea can be applied in the RAM model, by employing the point location data structure of [18] which offers the query time of $O((\log \log U)^2)$ in a $U \times U \times U$ integer grid.

Theorem 2. *Given n points in $U \times U \times U$ integer grid, dominance reporting queries can be answered in $O((\log \log U)^2 + k)$ worst case time in the RAM model using linear space.*

Reduction to the rank space and predecessor search are common techniques and tools which have appeared in many places (e.g., see [4] for more references and details) and they allow us obtain the following results.

Corollary 1. *For n points in $\mathbb{R}^3$, dominance reporting queries can be answered in $O((\log \log n)^2 + \log \log U + k)$ time using only linear space.*

Also, by using standard techniques to reduce orthogonal range reporting queries to dominance reporting queries (e.g., see [6, 8]) we can have the following.

Corollary 2. *There exists a data structure capable of answering 3D orthogonal range reporting queries on a $U \times U \times U$ grid in $O((\log \log n)^2 + \log \log U + k)$ time, using $O(n \log^3 n)$ space.*

The above can be extended to higher dimensions [4].

Corollary 3. *There exists a data structure capable of answering orthogonal range reporting queries in $\mathbb{R}^d$ using $O(n \log^{d+\varepsilon} n)$ space and with $O(\log^{d-3} n/(\log \log n)^{d-5} + k)$ query time.*

We also note that any improvements to the data structure for point location in a planar rectangular subdivision [18] can be carried over automatically to the dominance reporting problem and thus all the above corollaries.

Unfortunately, we cannot do the same trick to obtain an optimal algorithm in the external memory model, since in this model, up to our knowledge, there is no linear space algorithm with reasonable query time to combine with our approximate levels. Thus, to get an optimal algorithm in the external memory model, we need to develop additional tools and ideas. This is done in the next section.

5 The External Memory Model

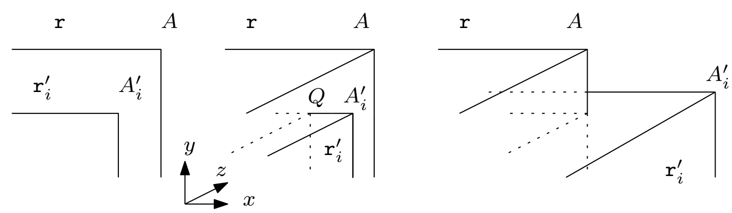
We use B to denote the block size in the external memory model. As we claimed in the introduction, our result on approximate levels can simplify the data structure of Vengroff and Vitter [12] by building a hierarchy of approximate levels. Of course, the space consumption would still be more than linear. To reduce the space complexity, we will need the following lemma.

Lemma 4. *Let $P \subset \mathbb{R}^3$ be a set of n points such that the level of each point is at most m for a parameter m. We can find $t = O(\frac{n}{m})$ sets, $V_1, \dots, V_t \subset P$, $|V_i| = O(m)$ such that for any downward corner $\mathtt{r}$ containing k points there exist $s = O(\frac{k}{m})$ sets $V_{t_1}, \dots, V_{t_s}$ with $|P_{\mathtt{r}} \setminus (\bigcup_{i=1}^{s} V_{t_i})| = O(m)$ in which $P_{\mathtt{r}} = P \cap \mathtt{r}$.*

Proof. Let $\mathcal{C} = \{\mathtt{r}_1, \dots, \mathtt{r}_t\}$ be an approximate Cm-level for a constant C to be determined later. With a slight abuse of the notation, we will use $\mathtt{r}_i$ to refer to both the downward corner $\mathtt{r}_i$ and the subset of P contained in $\mathtt{r}_i$. We claim $\mathtt{r}_1, \dots, \mathtt{r}_t$ are the sets claimed in the lemma. By Lemma 3 we know $t = O(\frac{n}{m})$.

Consider a downward corner $\mathtt{r}$ containing k points. According to Lemma 3, we can find an approximate m-level, $\mathcal{C}' = \{\mathtt{r}'_1, \dots, \mathtt{r}'_t\}$, of size $t' = O(\frac{k}{m})$ for the points inside $\mathtt{r}$. By definition, $\mathcal{C}'$ covers the $(\leq m)$-level of $P_{\mathtt{r}}$ and so every point of $P_{\mathtt{r}}$ is contained in at least one downward corner of $\mathcal{C}'$. Thus, $P_{\mathtt{r}} = \bigcup_{i=1}^{t'} \mathtt{r}'_i$. If we could show that for every $\mathtt{r}'_i \in \mathcal{C}'$ there is another downward corner $\mathtt{r}_j \in \mathcal{C}$ which contains $\mathtt{r}'_i$, then our lemma could be easily solved. Unfortunately this is not true and in fact, $\mathtt{r}'_i$ may contain $\Omega(n)$ points of P (although it can only contain $O(m)$ points of $P_{\mathtt{r}}$). Because of this we aim for a slightly weaker claim.

Let (x, y, z) be the coordinates of the apex A of $\mathtt{r}$ and (x', y', z') be the coordinates of the apex A'_i of $\mathtt{r}'_i$. By Lemma 3 we know each $\mathtt{r}'_i$ contains $O(m)$ points; assume the constant in the O notation is c. Pick $C > c$. We have four important cases:



(a) (b) (c)

Figure 2: (a) $\mathtt{r}'_i$ is contained inside $\mathtt{r}$. (b) Only the x-coordinate of A'_i is greater than that of A. (c) Two coordinates of A'_i are greater than those of A.

1. A dominates A'_i (Fig. 2(a)): In this case $\mathtt{r}'_i$ contains at most Cm points of P and is contained in at least one downward corner $\mathtt{r}_i \in \mathcal{C}$. Thus, in this case $\mathtt{r}'_i \subset \mathtt{r}_i$.
2. Only one coordinate of A'_i is not dominated by that of A (Fig. 2(b)): Without loss of generality assume it is the x-coordinate (i.e., $x < x', y > y'$ and $z > z'$). In this case, the point $Q = (x, y', z')$ is contained in $\mathtt{r}$ and thus dominates at most Cm points of P which means Q is contained in at least one downward corner $\mathtt{r}_i \in \mathcal{C}$. Thus, in this case $\mathtt{r}'_i \cap \mathtt{r} \subset \mathtt{r}_i$.
3. Only one coordinate of A dominates that of A'_i (Fig. 2(c)): This case can only happen for three elements of $\mathcal{C}'$, once for each coordinate; for instance, if $x' > x$ and $z' > z$, then $\mathtt{r}'_i$ contains at most Cm points with minimum y-coordinates in $P_{\mathtt{r}}$.
4. A'_i dominates A. This case can only happen if $P_{\mathtt{r}}$ contains less than Cm points.

For every downward corner $\mathtt{r}'_i \in \mathcal{C}'$, the first two cases provide us with another downward corner $\mathtt{r}_i \in \mathcal{C}$ such that $\mathtt{r}'_i \cap \mathtt{r} \subset \mathtt{r}_i$. The other two cases only cover $O(m)$ points. Thus, we can find at most t' downward corners $\mathtt{r}_1, \dots \mathtt{r}_{t''}, \in \mathcal{C}$ such that $|P_{\mathtt{r}} \setminus (\cup_{i=1}^{t''} \mathtt{r}_{t_i})| = O(m)$ with $t'' = O(k/m)$. □ □

Remarks. The closest theorem in the context of the halfspace range searching to the above lemma is the shallow partition theorem by Matoušek [15]; however, the above lemma does not partition the point set and does not cover all the points inside the downward corner $\mathtt{r}$. Also, it can be viewed as "output sensitive" in the sense that the number of sets contained or intersected by $\mathtt{r}$ depends on the number of points, k, contained in $\mathtt{r}$. It has been observed that such dependence on k is a desirable property [19]. Thus, an interesting question is whether it is possible to obtain a similar result for halfspace range searching; if so, we can also obtain an optimal halfspace range searching data structure in the external memory model.

Lemma 5. *There is a data structure for a set P of n points in $\mathbb{R}^3$ which can answer dominance reporting queries with $O(\log_B n + k/B)$ I/Os using $O(n)$ space.*

Proof. Partition P into subsets $P_1, \ldots, P_r$ in the following way: define P_1 to be the set of points $p \in P$ with level at most $B \log_B n$, remove P_1 and repeat and continue this operation until P is partitioned. This construction ensures that every point $p \in P_i$ has level at most $O(B \log_B n)$ in P_i.

Assume for every P_i we have a data structure which uses $O(|P_i|)$ space and can answer queries with $O(\log_B |P_i| + k/B)$ I/Os. Given a query $\mathtt{r}$, we start from P_1 and using the data structure implemented on P_1 we return all the points of P_1 inside $\mathtt{r}$ and then move on to the next set P_2 and continue this until we reach a point set P_i which does not contain any point in $\mathtt{r}$; at this point we terminate the search. The crucial observation is that if $\mathtt{r}$ contains at least one point from P_{i+1} then it must contain at least $B \log_B n$ points from P_{i-1}. This implies $k = \Omega(iB \log_B n)$ and thus the total query complexity will add up to $O((i+1) \log_B n + k/B) = O(\log_B n + k/B)$. In short, this means that it suffices to solve the problem for point sets P in which the level of every point is at most $B \log_B n$. This will be our assumption in the rest of the proof.

Let $m = B \log_B n$. Using Lemma 4, compute $t = O(\frac{n}{m})$ sets, $V_1, \ldots, V_t$, $|V_i| = O(m)$ and store the points of each set V_i sequentially. Consider a downward corner $\mathtt{r}$ containing k points. According to Lemma 4 there are $s = O(\frac{k}{m})$ sets $V_{t_1}, \ldots, V_{t_s}$ such that $|P_{\mathtt{r}} \setminus (\cup_{i=1}^{s} V_{t_i})| = O(m)$. We can represent the points inside $\mathtt{r}$ using $O(\frac{k}{m})$ pointers to sets V_{t_i} and an additional list of $O(m)$ points. This is our storage scheme for the list of points inside a downward corner $\mathtt{r}$.

To make the data structure, we build a hierarchy of approximate k_i-levels for $k_i = 2^i m, 0 \le i \le O(\log(n/B))$ and we store the list of points in every downward corner of the approximate levels using our storage scheme. Every downward corner in an approximate $2^i m$-level has $O(2^i m)$ points and thus will be stored using $O(2^i)$ pointers and a list of $O(m)$ points. Since this approximate level contains $O(\frac{n}{2^i m})$ ranges, the total space consumption for this level will be $O(\frac{n}{m} + m)$. Summing this up over all the approximate levels and including the space needed to store the sets $V_1, \ldots, V_t$ yields the space complexity of

$$O\left(n + \frac{n \log(n/B)}{m} + m \log(n/B)\right).$$

A simple calculation reveals this is always $O(n)$ for all values of B.

To answer the query, we find the smallest i that a downward corner $\mathtt{r}_j$ of an approximate $2^i m$-level contains $\mathtt{r}$. This can be done with $O(i+1)$ steps of point location, once for every approximate level up to the i-th one. This will also ensure that $\mathtt{r}_j$ contains $\Theta(m2^i) = \Theta(k)$ points. The output can be determined by a linear scan of all the points in $\mathtt{r}_j$; however, we have not stored the list of points of $\mathtt{r}_j$ directly and thus we must perform $O(\frac{k}{m})$ I/Os to just access the pointers, $O(\frac{k}{B})$ I/Os to access the list of points referenced by these pointers and finally an additional $O(\frac{m}{B})$ I/Os to access the list of points stored at $\mathtt{r}_j$. This amounts to $O\left((i+1) \log_B n + \frac{k+m}{B}\right) = O(\log_B n + k/B)$ I/Os. □ □

Combined with the standard reductions (e.g., see [6]), we can obtain the following corollary.

Corollary 4. *There is a data structure for a set of n points in $\mathbb{R}^3$ which that can answer orthogonal range reporting queries using $O(n \log^3 n)$ space and $O(\log_B n + k/B)$ I/Os.*

Using new point location data structure of Nekrich [13] we can also have the following result.

Corollary 5. *There is a data structure for a set of n points in $\mathbb{R}^3$ which that can answer dominance reporting queries using $O(n \log_B n)$ space and $O(\log \log_B U + (\log \log n)^2 + k/B)$ I/Os.*

The super-linear space complexity of the above corollary stems from the super-linear requirement of the point location data structure.

Acknowledgements. The author is in debt to Timothy Chan for many great suggestions, ideas and references which significantly improved the presentation of this article.

References

[1] Edelsbrunner, H., Overmars, M.H.: On the equivalence of some rectangle problems. Information Processing Letters **14** (May 1982)

[2] Agarwal, P.K.: Range searching. In Goodman, J.E., O'Rourke, J., eds.: CPC Handbook of Discrete and Computational Geometry. (2004)

[3] Agarwal, P.K., Erickson, J.: Geometric range searching and its relatives. In Chazelle, B., Goodman, J.E., Pollack, R., eds.: Advances in Discrete and Computational Geometry. AMS Press, Providence, RI (1999)

[4] Alstrup, S., Brodal, G.S., Rauhe, T.: New data structures for orthogonal range searching. In: FOCS '00: Proceedings of the 41st annual symposium on foundations of computer science, Washington, DC, USA, IEEE Computer Society (2000) 198

[5] Chazelle, B.: Functional approach to data structures and its use in multidimensional searching. SIAM J. Comput. **17**(3) (1988) 427–462

[6] Nekrich, Y.: A data structure for multi-dimensional range reporting. In: SCG '07: Proceedings of the 23rd annual symposium on computational geometry, New York, NY, USA, ACM (2007) 344–353

[7] Subramanian, S., Ramaswamy, S.: The P-range tree: a new data structure for range searching in secondary memory. In: SODA '95: Proceedings of the 6th annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA (1995) 378–387

[8] Vengroff, D.E., Vitter, J.S.: Efficient 3-D range searching in external memory. In: STOC '96: Proceedings of the 28th annual ACM symposium on theory of computing, New York, NY, USA, ACM (1996) 192–201

[9] Makris, C., Tsakalidis, A.: Algorithms for three-dimensional dominance searching in linear space. Inf. Process. Lett. **66**(6) (1998) 277–283

[10] Chazelle, B., Edelsbrunner, H.: Linear space data structures for two types of range search. Discrete and Computational Geometry **Volume 2**(1) (1987) 113,126

[11] JaJa, J., Mortensen, C.W., Shi, Q.: Space-efficient and fast algorithms for multidimensional dominance reporting and counting. In: 15th ISAAC. (2004) 558–568

[12] Vitter, J.S.: External memory algorithms and data structures: dealing with massive data. ACM Comput. Surv. **33**(2) (2001) 209–271 updated 2007 version at: http://www.cs.duke.edu/~jsv/Papers/catalog/.

[13] Nekrich, Y.: I/O-efficient point location in a set of rectangles. In: LATIN 2008: Theoretical Informatics. (2008) 687–698

[14] Agarwal, P.K., Efrat, A., Sharir, M.: Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. SIAM J. Comput. **29**(3) (2000) 912–953

[15] Matoušek, J.: Reporting points in halfspaces. Computational Geometry: Theory and Applications **2**(3) (1992) 169–186

[16] Agarwal, P.K., Arge, L., Erickson, J., Franciosa, P.G., Vitter, J.S.: Efficient searching with linear constraints. J. Comput. Syst. Sci. **61**(2) (2000) 194–216

[17] Ramos, E.A.: On range reporting, ray shooting and k-level construction. In: SCG '99:Proceedings of the 14th Annual Symposium on Computational Geometry. (1999) 390–399

[18] de Berg, M., van Kreveld, M., Snoeyink, J.: Two- and three-dimensional point location in rectangular subdivisions. J. Algorithms **18**(2) (1995) 256–277

[19] Aronov, B., Har-Peled, S., Sharir, M.: On approximate halfspace range counting and relative epsilon-approximations. In: SCG '07: Proceedings of the 23rd annual symposium on Computational geometry, New York, NY, USA, ACM (2007) 327–336