Optimal Halfspace Range Reporting in Three Dimensions

Peyman Afshani* Timothy M. Chan*

August 8, 2008

Abstract

We give the first optimal solution to a standard problem in computational geometry: three-dimensional halfspace range reporting. We show that n points in 3-d can be stored in a linear-space data structure so that all k points inside a query halfspace can be reported in $O(\log n + k)$ time. The data structure can be built in $O(n \log n)$ expected time. The previous methods with optimal query time requires superlinear ($O(n \log \log n)$) space.

We also mention consequences, for example, to higher dimensions and to external-memory data structures. As an aside, we partially answer another open question concerning the crossing number in Matoušek's *shallow partition theorem* in the 3-d case (a tool used in many known halfspace range reporting methods).

1 Introduction

In the *halfspace range reporting* problem, we want to preprocess a set of n points so that the points inside a query halfspace can be reported efficiently. This problem has a long and interesting history in computational geometry. The 2-d case was solved back in FOCS'83: Chazelle, Guibas, and Lee [12] were the first to obtain an $O(n)$-space data structure that can be built in $O(n \log n)$ time and can answer each query in $O(\log n + k)$ time, where k denotes the output size of the query. This result is clearly optimal under comparison-based models. A similarly optimal solution for the 3-d case, though, has remained elusive and is the subject of the present paper.

The 3-d problem is especially important. When the dimension is 4 or more, the consensus is that $O(\log n + k)$ query time is not possible with near linear space [18]. Furthermore, via a standard lifting transformation, the 3-d halfspace range reporting problem includes as special case *2-d circular range reporting*: preprocess n points in the plane so that the points inside a query circle, of any radius, can be reported quickly. In turn, this problem is closely linked to *2-d k-nearest-neighbors search*: preprocess n points in the plane so that the k exact nearest neighbors to a query point, for any k, can be reported in arbitrary order quickly, where distances are measured in the Euclidean metric. Both 2-d problems are among the most natural proximity problems imaginable, and have been studied extensively since the early days of computational geometry [19, 21].

Table 1 summarizes previous work. The results in the early entries are perhaps portrayed a little unfairly, because basic knowledge in the field was still being developed at the time; for example, Bentley and Maurer's space requirement automatically improves to $O(n^3)$ by known combinatorial bounds on order-k Voronoi diagrams, and Chazelle and Preparata's space requirement is actually $O(n \log^2 n \log \log n)$ by Clarkson and Shor's bounds on $(\leq k)$-sets [14]. By the early 90s, with the work by Matoušek [18] and its precursors,

*School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada ({pafshani,tmchan}@uwaterloo.ca).
Work supported by NSERC.

	space	query time	preprocessing time
(*) Bentley, Maurer'79 [7]	n^4	$\log n \log\log n + k$	
Cole, Yap (FOCS'83) [15]	n^4	$\log n + k$	
(*) Chazelle *et al.*'86 [11]	$n \log^2 n \log^2 \log n$	$\log n + k$	$n \log^5 n \log^2 \log n$
	$n \log^2 n$	$\log n + k$	
	$n \log n$	$k \log^2 n$	$n \log n$
Chazelle, Preparata (SoCG'85) [13]	$n \log^8 n \log^4 \log n$	$\log n + k$	
Aggarwal *et al.* (STOC'90) [4]	$n \log n$	$\log n + k$	$n^3 \log n$
	$n \log n$	$\log n + k$	$n \log^2 n \log\log n$ mc.
Matoušek (FOCS'91) [18]	$n \log\log n$	$(\log n)^{O(\log\log n)} + k$	$n \log n$
	n	$n^{\varepsilon} + k$	$n \log n$
Chan (FOCS'98) [8]	$n \log n$ expected	$\log n + k$ expected	$n \log n$ expected
	$n \log\log n$	$\log n + k$	
Ramos (SoCG'99) [20]	$n \log\log n$	$\log n + k$	$n \log n$ expected

Table 1: Past results for 3-d halfspace range reporting. Those marked (*) are specifically for 2-d circular range reporting or k-nearest-neighbors search. ("mc." stands for Monte Carlo.)

powerful tools such as *cutting lemmas* and *partition theorems* have been identified to tackle range searching problems in general, even in higher dimensions [3]; in fact, the above table entries on Matoušek's results were not explicitly stated in his paper [18], which focused on halfspace range reporting in dimensions larger than 3. However, despite these advances, for 3-d halfspace range reporting or 2-d circular range reporting, the current best data structures with $O(\log n + k)$ query time require superlinear $O(n \log\log n)$ space, as found by Chan [8] or Ramos [20] or P. Agarwal (personal communication).

In this paper, we end the nearly-30-year-old quest for an optimal data structure for 3-d halfspace range reporting or 2-d circular range reporting or 2-d k-nearest-neighbors search: Section 2 unveils an $O(n)$-space data structure that supports queries in $O(\log n + k)$ worst-case time. Furthermore, the structure can be preprocessed by a randomized algorithm that runs in $O(n \log n)$ expected time. Surprisingly (and somewhat disappointingly), our solution does not require invention of any new tools at all, not even modifications of old tools. We apply a combination of Matoušek's *shallow cutting lemma* and *shallow partition theorem* [18], which were already used in previous methods [8, 20]. The way we generate a partition tree using the shallow partition theorem is even simpler than Matoušek's! The key innovation lies in the analysis, and "merely" involves solving a (slightly nonobvious) recurrence.

In Section 3, we briefly discuss other implications of the idea, e.g., to higher dimensions and to external-memory data structures. In particular, an optimal external-memory method for 3-d halfspace range reporting remains open, although we can now get within a very small, iterated-logarithmic factor to optimal.

Finally in Section 4, we turn to a separate combinatorial question concerning Matoušek's shallow partition theorem in the 3-d case (see Section 2 for the statement). The theorem establishes an $O(r^{1-\lfloor d/2 \rfloor})$ upper bound on the crossing number for dimensions $d \geq 4$, and an $O(\log r)$ upper bound for the cases $d = 2, 3$. After noting the optimality of the bound for $d \geq 4$, Matoušek wrote: "For $d = 2, 3$, it would be interesting to see to what extent the crossing number can be improved. It is not clear whether a crossing number bounded by a constant can be attained (we conjecture that it cannot)." We confirm this conjecture for $d = 3$, by exhibiting an example where the crossing number is $\Omega(\log r / \log\log r)$, proving that the upper bound is almost tight. Rather unexpectedly, the same proof answers another open question of Arge *et al.* [5] on a seemingly different topic: 2-d 3-sided orthogonal range searching in external memory "without redundancy".

2 The Optimal Solution

Our method uses the shallow cutting lemma and the shallow partition theorem as black boxes, but is otherwise essentially self-contained. We begin by reviewing these two tools.

By duality, the 3-d halfspace range reporting problem is equivalent to preprocessing a set H of n planes in 3-d with the goal to report all planes below a query point. The number of planes below q is called the *level* of q. Intuitively, a point is "shallow" if its level is small. The closure of the set of all points with level at most t is known as the $(\leq t)$*-level* of H. The shallow cutting lemma is the following:

Shallow Cutting Lemma. *Given a set H of n planes in 3-d and a parameter $t \leq n$, there exists a set of $O(n/t)$ disjoint vertical prisms (tetrahedra with a vertex at $(0,0,-\infty)$) which cover the $(\leq t)$-level of H, such that each prism intersects $O(t)$ planes.*

This set of prisms together with the list of planes crossing each prism, for every $t = 1,2,4,8,\ldots$, can be constructed in $O(n \log n)$ total expected time.

Matoušek [18] proved the existence of such a cutting (in fact, in a slightly more general form). Ramos [20] contributed the $O(n \log n)$ bound on the randomized construction time, and Chan [8] observed that the cutting can be assumed to be a set of vertical prisms. The shallow cutting lemma alone already implies efficient methods for halfspace range reporting, as shown below. Although the following proposition has not been stated explicitly before, the idea is implicitly known (in [8], and to some extent, in [20]).

Proposition 2.1 *Given a data structure A for the 3-d halfspace range reporting problem with $O(n)$ space and $O(f(n)+k)$ query time, it is possible to build another data structure A' with $O(\ell n)$ space and $O(\log n + f^{(\ell)}(n) + k)$ query time, assuming that $f(n) \leq n/2$ is a decreasing function satisfying $f(O(n)) = O(f(n))$.*

Proof: W.l.o.g., assume $f^{(\ell-1)}(n) > \log n$ (because otherwise, we may decrement ℓ). We use the shallow cutting lemma with parameter $t = f^{(i)}(n)$ to build a set C_i of $O(n/f^{(i)}(n))$ prisms, for each $i = 0,\ldots,\ell-1$. Implement the data structure A on the $O(f^{(i)}(n))$ planes crossing each prism. The total stoarge cost is $O(\sum_i (n/f^{(i)}(n)) \cdot f^{(i)}(n)) = O(\ell n)$.

Let q be the query point, with k planes below q. We can search for a prism in C_i that contains q, for a given i, by planar point location [19]. Find the prism $\Delta \in C_i$ that contains q with the largest i. Note that $k > f^{(i+1)}(n)$ if $i \neq \ell - 1$ (because otherwise i would not be the largest). Any plane below q must cross Δ, and thus the query can be answered by using the data structure A implemented on the $O(f^{(i)}(n))$ planes crossing Δ. The query time is $O(f(f^{(i)}(n)) + k) = O(f^{(i+1)}(n) + k)$, which is $O(k)$ if $i \neq \ell - 1$ or $O(f^{(\ell)}(n) + k)$ otherwise.

In addition, the search for i requires $\ell - i$ point location steps, which can be done in $O((\ell - i)\log n)$ time. This cost never exceeds $O(\log n + k)$, since $\ell - i$ is at most $O(1 + \log \left\lceil f^{(i+1)}(n)/f^{(\ell-1)}(n) \right\rceil) = O(1 + \log\lceil k/\log n \rceil)$. □

For example, using the trivial bound $f(n) = n/2$, we can set $\ell = \log n$ and immediately get $O(n \log n)$ space and $O(\log n + k)$ query time. By using instead a known linear-space data structure with sublinear query time $f(n) \leq n^{1-\varepsilon}$ (such as [17]), we can set $\ell = \Theta(\log\log n)$ and recover the previous result of $O(n \log\log n)$ space and $O(\log n + k)$ query time.

Our second ingredient is the shallow partition theorem, also proved by Matoušek [18] (who referred to it more appropriately, but less concisely, as the "partition theorem for shallow (hyper)planes"). This time, we return to the primal setting. Intuitively, a halfspace, or its defining plane, is "shallow" if the halfspace contains a small number of data points.

Shallow Partition Theorem. *For a set P of n points in 3-d and a parameter $r \leq n$, there exists a partition of P into $O(r)$ subsets, each of size $\Theta(n/r)$, and each enclosed by a tetrahedron, such that any halfspace which has at most n/r points of P crosses $O(\log r)$ tetrahedra.*

If $r \leq n^\varepsilon$ for a sufficiently small constant $\varepsilon > 0$, the partition and tetrahedra can be constructed in $O(n \log r)$ time.

(Note that here it does not matter whether the crossing number counts tetrahedra intersecting the halfspace or tetrahedra intersecting the halfspace's boundary, since a halfspace with at most n/r points can strictly contain $O(1)$ tetrahedra.)

The proof of the above theorem actually uses the shallow cutting lemma. Matoušek applied the theorem recursively with $r = n^\varepsilon$ to get a data structure for halfspace range reporting. The *partition tree* thus generated has $O(\log\log n)$ levels, resulting in $O(n \log\log n)$ space and $O((\log n)^{O(\log\log n)} + k)$ query time in 3-d. Our first, actual new idea is an improvement of the space bound of this data structure to linear.

Proposition 2.2 *We can design a data structure of $O(n)$ space which can answer 3-d halfspace range reporting queries in $O((\log n)^{O(\log\log n)} + k)$ time.*

Proof: We pick a sufficiently small constant $\varepsilon > 0$. We use the shallow partition theorem with parameter $r = n^\varepsilon$ to build subsets P_i and corresponding tetrahedra. We recursively build the data structure for each P_i. In addition, we store the vertices of the $O(r)$ tetrahedra in some trivial halfspace range reporting data structure A_0 with $r^{O(1)}$ space and $O(\log r + k)$ query time. We stop the recursion when the number of points drops below a constant. Note that the points are only stored only at the leaves of the resulting tree. For a sufficiently large constant c, the total space satisfies the following recurrence

$$S(n) \;=\; \sum_{i=1}^{cn^\varepsilon} S(n_i) + n^{O(\varepsilon)} \qquad \text{where } \sum_i n_i = n \text{ and } \max_i n_i \leq cn^{1-\varepsilon},$$

which solves to $O(n)$.

Let h be the query halfspace, containing k points. We first find up to $\kappa := c\log r$ tetrahedra which are crossed by h, for a sufficiently large c. Since a halfspace crossing a tetrahedron must contain one of its vertices, this can be done in $O(\log r + \kappa)$ time by A_0. Two cases are possible, depending on whether the actual number of crossed tetrahedra is smaller or larger than κ. The former, "small" case is simple: we just recursively answer the query for the subsets corresponding to the crossed tetrahedra. In the latter, "large" case, h cannot be shallow, i.e, h contains more than $n/r = n^{1-\varepsilon}$ points. As the output size k is close to linear, Matoušek observed that here we can afford to switch to any of the known linear-space data structure with sublinear query time (such as [17]), since the query cost can be absorbed under the $O(k)$ term. However, this approach requires augmenting the tree with secondary structures at every node, and consequently the space bound increases to $O(n \log\log n)$.

Ramos [20] suggested a different idea to handle the large case, without secondary data structures: namely, find a strengthened version of the partition theorem that can produce a single partition with simultaneously good crossing number for shallow halfspaces and sublinear crossing number for deep halfspaces. Unfortunately, he was only able to obtain such a theorem for even dimensions.

We suggest a new idea to handle the large case, without secondary data structures and without a new partition theorem: namely, just recurse in all $O(r)$ subsets! It is a surprise that this practically trivial idea was overlooked, though it is not as easy to see why the analysis still works out…

Denote by $Q(n,k)$ the query time spent on a subset of size n with output size k. We have the following, somewhat unusual recurrence, for some constant c':

$$Q(n,k) \leq \begin{cases} \sum_{i=1}^{c\log n} Q(cn^{1-\varepsilon}, k_i) + c'\log n & \text{where } k \;=\; \sum_i k_i \;\leq\; n^{1-\varepsilon} \\ \sum_{i=1}^{cn^\varepsilon} Q(cn^{1-\varepsilon}, k_i) + c'\log n & \text{where } k \;=\; \sum_i k_i \;>\; n^{1-\varepsilon}. \end{cases} \tag{1}$$

We guess that the solution of the recurrence obeys $Q(n,k) \le f(n) + kg(n)$, where f and g are some functions to be determined, satisfying $f(n) \ge f(cn^{1-\varepsilon})$ and $g(n) \ge g(cn^{1-\varepsilon})$.

First we verify the claim by induction for the case when $k \le n^{1-\varepsilon}$:

$$\begin{aligned} Q(n,k) &\le \sum_{i=1}^{c\log n} [f(cn^{1-\varepsilon}) + k_i g(cn^{1-\varepsilon})] \,+\, c'\log n \\ &\le f(n) + kg(n), \end{aligned}$$

if we choose a function $f(n)$ that is defined by the recurrence

$$f(n) \,:=\, (c\log n)\, f(cn^{1-\varepsilon}) \,+\, c'\log n.$$

For the case when $k > n^{1-\varepsilon}$, we have

$$\begin{aligned} Q(n,k) &\le \sum_{i=1}^{cn^{\varepsilon}} [f(cn^{1-\varepsilon}) + k_i g(cn^{1-\varepsilon})] \,+\, c'\log n \\ &\le cn^{\varepsilon} f(n) + kg(cn^{1-\varepsilon}) + c'\log n \\ &\le k\cdot[cf(n)/n^{1-2\varepsilon} + g(cn^{1-\varepsilon}) + (c'\log n)/n^{1-\varepsilon}] \qquad \text{since } k > n^{1-\varepsilon} \\ &< f(n) + kg(n), \end{aligned}$$

if we choose a function $g(n)$ that is defined by the recurrence

$$g(n) \,:=\, g(cn^{1-\varepsilon}) + cf(n)/n^{1-2\varepsilon} + (c'\log n)/n^{1-\varepsilon}.$$

Having shown that $Q(n,k) \le f(n) + kg(n)$ for these choices of $f(n)$ and $g(n)$, we just need to bound these two functions. The f recurrence easily solves to $f(n) = (\log n)^{O(\log\log n)}$. The g recurrence, after expansion, yields a sum whose terms converge to 0 superexponentially. Thus $g(n) = O(1)$, fortunately, and we are done. $\square$

For our final maneuver, we invoke the reduction in Proposition 2.1. While the query time in Proposition 2.2 might initially look weak, we only need $\ell = 2$ iterations of Proposition 2.1 to bring the bound to optimality: for $f(n) := (\log n)^{O(\log\log n)} = 2^{O(\log^2 \log n)}$, we have $f(f(n)) = 2^{O(\log^2 \log\log n)}$, which is already sublogarithmic!

Theorem 2.3 *We can design a data structure of $O(n)$ space which can answer 3-d halfspace range reporting queries in $O(\log n + k)$ time.*

From the stated time bounds in the shallow partition theorem and shallow cutting lemma, it is straightforward to see that the preprocessing time for the data structure in Proposition 2.2 is $O(n\log n)$, and it remains unchanged in the application of Proposition 2.1. The final preprocessing algorithm is randomized only because of Ramos' shallow cutting subroutine.

3 Consequences/Applications

k-lowest-planes queries. Consider the following related problem: preprocess a set H of n planes in 3-d so that we can report the k lowest planes at a query vertical line in arbitrary order, for any k. The 2-d k-nearest-neighbors search problem reduces to this problem by the standard lifting transformation. We can solve the problem with the same bounds, but additional effort is required, since the method in Proposition 2.2 cannot be easily modified.

Theorem 3.1 *We can design a data structure of $O(n)$ space which can answer 3-d k-lowest-planes queries in $O(\log n + k)$ time. The expected preprocessing time is $O(n \log n)$.*

Proof: We use the shallow cutting lemma with parameter $t = 2^i$ to build a set C_i of $O(n/2^i)$ prisms, for each $i = 0, \ldots, \log n$. The total size of these sets is $O(\sum_i (n/2^i)) = O(n)$.

Given a query vertical line ℓ and a number k, pick i with $2^{i-1} < k \leq 2^i$ and find the prism $\Delta \in C_i$ that intersects ℓ, by planar point location in $O(\log n)$ time. Let q be the intersection of the boundary of Δ with ℓ. All k lowest planes at ℓ must lie below q, and thus the query can be answered by using the data structure in Theorem 2.3 (in the dual) to find the planes below q, and then selecting the k lowest among these planes. Since there are $O(2^i) = O(k)$ planes below q, the additional time is $O(k)$. □

Higher dimensions. For any constant $d \geq 4$, Matoušek's partition tree [18] yields an $O(n \log\log n)$-space data structure that answers d-dimensional halfspace range reporting queries in $O(n^{1-1/\lfloor d/2 \rfloor} \operatorname{polylog} n + k)$ time. Ramos [20] managed to reduce the space to linear for even d with considerable effort. We get linear space with a much simpler approach, for all constant dimensions, even and odd.

Theorem 3.2 *We can design a data structure of $O(n)$ space which can answer d-dimensional halfspace range reporting queries $O(n^{1-1/\lfloor d/2 \rfloor} \operatorname{polylog} n + k)$. The expected preprocessing time is $O(n \log n)$.*

Proof: We use the same approach as in Proposition 2.2, with the higher-dimensional version of the shallow partition theorem. The recurrence (1) now changes to

$$Q(n,k) \leq \begin{cases} \sum_{i=1}^{O(n^{\varepsilon(1-\lfloor d/2 \rfloor)})} Q(cn^{1-\varepsilon}, k_i) + O(n^\varepsilon) & \text{where } k = \sum_i k_i \leq n^{1-\varepsilon} \\ \sum_{i=1}^{O(n^\varepsilon)} Q(cn^{1-\varepsilon}, k_i) + O(n^\varepsilon) & \text{where } k = \sum_i k_i > n^{1-\varepsilon}. \end{cases} \tag{2}$$

This time, $f(n) = O(n^{1-\lfloor d/2 \rfloor} \operatorname{polylog} n)$, while the g recurrence still solves to $g(n) = O(1)$ if ε is sufficiently small. □

External memory. Agarwal *et al.* [2] investigated the 3-d halfspace range reporting problem in the external memory model. To be consistent with the literature, let N and K denote the size of the given point set and the output size of a query, instead of n and k. Let B denote the block size. Agarwal *et al.* adapted Chan's method [8] to obtain a data structure that occupies $O((N/B) \log N)$ blocks and answers queries in $O(\log_B N + K/B)$ I/Os. We get the following improvement:

Theorem 3.3 *We can design an external-memory data structure with $O((N/B) \log^*(N/B))$ blocks of space and $O(\log_B N + K/B)$ I/Os per query.*

Proof: First we adapt the data structure in Proposition 2.2. We build a partition tree with parameter $r = (N/B)^\varepsilon$, which results in $O(r)$ sets of size $\Theta(N/r) = \Theta(B(N/B)^{1-\varepsilon})$. We stop the recursion when the number of points drops to $\Theta(B)$, in which case the points are stored sequentially in $O(1)$ blocks. The recurrence for the query I/O cost turns out to be identical to (1), if we simply make a change of variables: $n = N/B$ and $k = K/B$. Consequently, we get a method with $O(N/B)$ blocks of space and $O((\log(N/B))^{O(\log\log(N/B))} + K/B)$ I/Os per query.

To lower query time, we adapt the reduction from Proposition 2.1. The same analysis holds if we set $n = N/B$ and $k = K/B$, and replace all occurrences of "$\log n$" with "$\log_B N$", since a planar point location query now has $O(\log_B N)$ cost [16]. Consequently, we get a method with $O(\ell N/B)$ blocks of space and $O(\log_B N + 2^{O(\log^{(\ell)}(N/B))} + K/B)$ I/Os per query. We can set $\ell = \log^*(N/B)$. □

We leave open an interesting question: can the $\log^*$ factor be eliminated? Apart from algorithmic significance of the external memory model, the above has a purely combinatorial consequence:

Corollary 3.4 *For a set P of n points in 3-d and a parameter $r \leq n$, there exists a cover of P by $O(r \log^* r)$ subsets, each of size $O(n/r)$, such that any halfspace which contains k points of P can be covered by $O(\log_{n/r} n + kr/n)$ subsets.*

Note the resemblance of this corollary to the shallow partition theorem. The main differences are that a point may appear in multiple subsets and the crossing/covering number bound should be valid for all k, not just for the shallow case with $k \leq n/r$. (See [6] for more on the latter issue.) With a moment's thought, one can see that finding such a combinatorial "cover" theorem is roughly equivalent to finding an efficient external memory data structure.

Miscellaneous.

- We can also get a result for 3-d halfspace range reporting or 2-d circular range reporting on the word RAM model, where we assume that the input points have integer coordinates from $\{1, \ldots, U\}$ and that the word size is at least $\log U$. By using known word RAM results [10] for the point location queries in Proposition 2.1, we immediately get an $O(n)$-space data structure with query time $O(\min\{\log n/\log\log n,\ \sqrt{\log U/\log\log U}\} + k)$.
- Chan [9] gave the first polylogarithmic dynamic data structure for 2-d exact nearest neighbor search and for 3-d convex hull queries. This method can be implemented with $O(n \log\log n)$ space, as was reported in the full version of that paper, by using 3-d halfspace range reporting. We can now reduce the space of this dynamic data structure to $O(n)$.
- Afshani [1] recently gave a data structure for the *3-d dominance reporting* problem with $O(n)$ space and $O(\log n + k)$ query time in the pointer machine model. Our method here can be adapted to solve this problem as well, but 3-d dominance reporting is in a way simpler than 3-d halfspace reporting. Indeed, his method only requires a version of Proposition 2.1 with $\ell = 2$ and does not need a partition-tree approach like Proposition 2.2, since a method with $O(n)$ space and $O(\text{polylog}\, n + k)$ query time was already previously known for dominance. Further, Afshani exploited properties specific to dominance and obtained an optimal external memory result, which we are currently unable to get for halfspace reporting without the $\log^*$ factor.

4 A Lower Bound on the 3-d Shallow Partition Theorem

In this final section, we investigate a related but independent question, raised by Matoušek [18]: is the $O(\log r)$ crossing number bound in the 3-d shallow partition theorem optimal? Besides being a natural combinatorial question, lowering the crossing number to $O(1)$ would have algorithmic implications, for example, to potentially improving our external-memory result for 3-d halfspace range reporting, or to improving the relative (p, ε)-approximation results of Aronov *et al.* [6]. We show, however, that $O(1)$ is not possible.

To improve readability of the proof, we first simplify the geometry by considering a special type of ranges in the 2-d setting: 3-sided rectangles, i.e., rectangles of the form $[a, b) \times [c, \infty)$. We show afterwards how the proof for halfspaces in 3-d would follow.

Lemma 4.1 *For any $r \leq n^{1-\varepsilon}$ for a fixed $\varepsilon > 0$, there exists a set P of n points in 2-d such that for any partition of P into $O(r)$ subsets of size $O(n/r)$, some 3-sided rectangle contains $O(n/r)$ points and contains points from $\Omega(\log r/\log\log r)$ different subsets.*

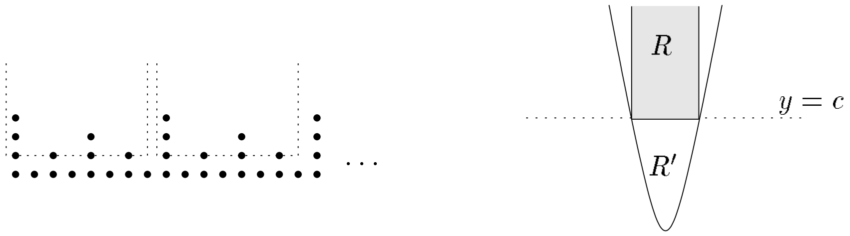

Figure 1: (Left) Proof of Lemma 4.1: the point set P and two level-1 rectangles (with $b = 2$ and $k = 4$). (Right) Proof of Theorem 4.2: the rectangle $R = [a, b) \times [c, \infty)$ and the parabolic range $R' = \{(x, y) : y - c \geq m(x - a)(x - b)\}$. As $m \to \infty$, R' approaches the slab $[a, b] \times \mathbb{R}$.

Proof: Let $k = n/r$. Fix a parameter $b \geq 2$. Let $\ell = \min\{\log_b k, \log_b(n/k)\}$. Our point set P is simple: for each $i = 0, \ldots, \ell$, we just place about n/b^i uniformly spaced points of distance b^i apart on the horizontal segment $[0, n) \times \{i\}$. (See Figure 1 (left).) This set has $\Theta(n + n/b + \cdots) = \Theta(n)$ points.

As it turns out, we only need to focus on *basic* 3-sided rectangles of the form $[(j-1)kb^i, jkb^i) \times [i, \infty)$ for $i = 0, \ldots, \ell$ and $j = 1, \ldots, \frac{n}{kb^i}$. Call such a rectangle a *level-i* rectangle. Notice that each such basic rectangle contains $\Theta(k + k/b + \cdots) = \Theta(k)$ points.

Suppose we are given a partition of P into $O(n/k)$ subsets of size $O(k)$. Think of points in the same subset as having the same color. We want to find a basic rectangle that contains $\Omega(\ell)$ colors.

If two points p and q are in a common level-0 rectangle and have the same color, and p is strictly below q, then we say that p is *killed* by q.

If some level-$(i+1)$ rectangle contains $\Omega(\ell)$ colors, we can stop. Otherwise, the points in one level-$(i+1)$ rectangle can kill at most $O(\ell k)$ points, since there are at most $O(k)$ points of each color. Therefore, the points at or above $y = i + 1$ can kill a total of at most $O\left(\frac{n}{kb^{i+1}} \cdot \ell k\right) = O(\ell n/b^{i+1})$ points. In particular, the number of points at $y = i$ that are killed is at most $O(\ell n/b^{i+1})$.

We say that a level-0 rectangle R is *bad* if for some i, all points in R at $y = i$ are killed. Since a level-0 rectangle contains $\Theta(k/b^i)$ points at $y = i$, the number of bad level-0 rectangles is at most $\sum_{i=0}^{\ell} O\left(\frac{\ell n/b^{i+1}}{k/b^i}\right) = O\left(\ell \cdot \frac{\ell n}{bk}\right)$. By choosing $b \approx \min\{\log^2 k, \log^2(n/k)\}$, the bound can be made smaller than n/k, the number of level-0 rectangles.

Thus, some level-0 rectangle R is not bad. By picking one point in R at each horizontal line that is not killed, we then get $\Omega(\ell)$ points of different colors in R. The theorem follows since $\ell = \Omega(\min\{\log k / \log\log k, \log(n/k)/\log\log(n/k)\})$ for our choice of b. $\square$

Theorem 4.2 *For any $r \leq n^{1-\varepsilon}$ for a fixed $\varepsilon > 0$, there exists a set P of n points in 3-d such that for any partition of P into $O(r)$ subsets of size $O(n/r)$, some halfspace contains $O(n/r)$ points and contains points from $\Omega(\log r/\log\log r)$ different subsets.*

Proof: We show how to embed the 2-d point set from Lemma 4.1 in 3-d, in such a way that each 3-sided rectangle in 2-d is *realizable* by a halfspace in 3-d, i.e., the subset of points inside the rectangle corresponds precisely to the subset of points inside the halfspace.

All our 3-d points will lie on $z = x^2$. (Any strictly convex curve in the xz-plane will do, but this particular choice allows us to specify the halfspaces more explicitly.) Initially, we simply lift the 2-d points (x, y) to (x, y, x^2). We will later modify the y-coordinates of the points repeatedly, but the y-coordinates will only shift downward and the relative y-ordering of the points will remain the same. We may assume that the x-coordinates of the 2-d points are all distinct.

The number of combinatorially different rectangles is finite. We proceed from top to bottom, sweeping the xy-plane with a horizontal sweep line $y = c$. We maintain the invariant that the y-coordinates of all points above $y = c$ have been fixed, and that all 3-sided rectangles above $y = c$ have been realized. Suppose the sweep line encounters the bottom edge of a 3-sided rectangle $R = [a, b) \times [c, \infty)$. We take the halfspace $z \leq x^2 - (x - a)(x - b) + (1/m)(y - c)$ for some value $m > 0$. Note that the inequality is indeed linear because of the cancellation of the x^2 term. The intersection of this halfspace with $z = x^2$ projects to a parabolic range R' in the xy-plane: $y - c \geq m(x - a)(x - b)$. By choosing m sufficiently large, we can ensure that R' and R contain the same subset of points above $y = c$. (See Figure 1 (right).) By shifting all points below $y = c$ sufficiently downward to lie beneath the y-minimum of the parabola, we can ensure that R' contains no points below $y = c$. Thus, R has been successfully realized. $\square$

We remark that the $\Omega(\log r/ \log \log r)$ bound is tight for the particular family of point sets considered in the proof of Lemma 4.1 (even if we set $b = 2$).

Incidentally, Lemma 4.1 resolves an "interesting open problem" from a paper by Arge *et al.* [5, second paragraph in Section 2.2]. Specifically, they considered 3-sided orthogonal range searching in the plane in external memory, and asked whether there exists an indexing scheme with *redundancy* 1 and *access overhead* $O(1)$. Our lemma implies that such a scheme is impossible.

References

[1] P. Afshani. On dominance reporting in 3D. To appear in *Proc. 16th European Sympos. Algorithms*, 2008.

[2] P. K. Agarwal, L. Arge, J. Erickson, P.G. Franciosa, and J. S. Vitter. Efficient searching with linear constraints. *J. Comput. Sys. Sci.*, 61:192–216, 2000.

[3] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In *Discrete and Computational Geometry: Ten Years Later* (B. Chazelle, J. E. Goodman, and R. Pollack, ed.), AMS Press, pages 1–56, 1999.

[4] A. Aggarwal, M. Hansen, and T. Leighton. Solving query-retrieval problems by compacting Voronoi diagrams. In *Proc. 22nd ACM Sympos. Theory Comput.*, pages 331–340, 1990.

[5] L. Arge, V. Samoladas, and J. S. Vitter. On two-dimensional indexability and optimal range search indexing. In *18th ACM Sympos. Principles of Database Systems*, pages 346–357, 1999.

[6] B. Aronov, S. Har-Peled, and M. Sharir. On approximate halfspace range counting and relative ε-approximations. In *Proc. 23rd ACM Sympos. Comput. Geom.*, pages 327–336, 2007.

[7] J. L. Bentley and H. A. Maurer. A note on Euclidean near neighbor searching in the plane. *Inform. Process. Lett.*, 8:133–136, 1979.

[8] T. M. Chan. Random sampling, halfspace range reporting, and construction of $(\leq k)$-levels in three dimensions. *SIAM J. Comput.*, 30:561–575, 2000.

[9] T. M. Chan. A dynamic data structure for 3-d convex hulls and 2-d nearest neighbor queries. In *Proc. 17th ACM-SIAM Sympos. Discrete Algorithms*, pages 1196–1202, 2006. Full version at http://www.cs.uwaterloo.ca/ ~tmchan/dch3d_9_06.ps

[10] T. M. Chan and M. Pătraşcu. Point location in sublogarithmic time and other transdichotomous results in computational geometry. Submitted to *SIAM J. Comput.* Preliminary versions in *Proc. 47th IEEE Sympos. Found. Comput. Sci.*, pages 325–332 and 333–342, 2006.

[11] B. Chazelle, R. Cole, F. P. Preparata, and C. K. Yap. New upper bounds for neighbor searching. *Inform. Control*, 68:105–124, 1986.

[12] B. Chazelle, L. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25:76–90, 1985.

[13] B. Chazelle and F. P. Preparata. Halfspace range search: an algorithmic application of k-sets. *Discrete Comput. Geom.*, 1:3–93, 1986.

[14] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[15] R. Cole and C. K. Yap. Geometric retrieval problems. *Inform. Control*, 63:39–57, 1984.

[16] M. T. Goodrich, J.-J. Tsay, D. E. Vengroff, and J. S. Vitter. External-memory computational geometry. In *Proc. 34th IEEE Sympos. Found. Comput. Sci.*, pages 714–723, 1993.

[17] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.

[18] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2:169–186, 1992.

[19] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[20] E. Ramos. On range reporting, ray shooting, and k-level construction. In *Proc. 15th ACM Sympos. Comput. Geom.*, pages 390–399, 1999.

[21] M. I. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th IEEE Sympos. Found. Comput. Sci.*, pages 151–162, 1977.