Counting Colors in Boxes *

Haim Kaplan† Natan Rubin‡ Micha Sharir§ Elad Verbin¶

Abstract

Let P be a set of n points in $\mathbb{R}^d$, so that each point is colored by one of C given colors. We present algorithms for preprocessing P into a data structure that efficiently supports queries of the form: Given an axis-parallel box Q, count the number of distinct colors of the points of $P \cap Q$. We present a general and relatively simple solution that has polylogarithmic query time and worst-case storage about $O(n^d)$. It is based on several interesting structural properties of the problem that we derive. We also show that for random inputs, the data structure requires almost linear expected storage.

We then present several techniques for achieving space-time tradeoff. In $\mathbb{R}^2$, the most efficient solution uses fast matrix multiplication in the preprocessing stage. In higher dimensions we obtain a tradeoff using simpler mechanisms. We give a reduction from matrix multiplication to the offline version of problem, which shows that in $\mathbb{R}^2$ our time-space tradeoffs are close to optimal in the sense that improving them substantially would improve the best exponent of matrix multiplication. Finally, we present a generalized matrix multiplication problem and show its intimate relation to counting colors in boxes in any dimension.

1 Introduction

We consider the following range counting problem. Let P be an input set of n points in $\mathbb{R}^d$, each colored in one of C different colors. Our goal is to preprocess P into a data structure that, for a given query axis-parallel box $Q \subset \mathbb{R}^d$, can efficiently count the number of *distinct colors* of points in $Q \cap P$. We call this problem *colored orthogonal range counting*. In this work we only deal with the static setting of the problem, not allowing insertions or deletions to the data structure.

The problem arises in many applications. For example, in database applications, the items in the database have some attribute (e.g., the city where they have been born, the college that they have attended, or even numerical attributes such as their age), and the query asks for the number of different *attributes* of all the items in a query box (e.g., how many different cities of birth do the persons in a query box have?). In geometric contexts, the problem arises, e.g., in the following scenario: We are given C rectilinear polygons in the plane with a total of n edges, and wish to count the number of distinct polygons that intersect a given query box. Similar problems arise in higher dimensions.

Related work. Gupta *et al.* (see the recent survey [12]) have initiated the study of *colored intersection range searching*, where one is given n colored objects of constant description complexity, and wishes to construct a data structure that can report or count the colors of objects in a query range. The results that are most relevant to our study are: (a) In $\mathbb{R}^1$, colored range counting can be done with $O(\log n)$ query time, $O(n)$ storage and $O(n \log n)$ preprocessing [4, 16]. (b) In $\mathbb{R}^2$, colored range counting can be done with $O(\log^2 n)$ query time, $O(n^2 \log^2 n)$ storage and preprocessing [16]. As has been observed, colored variants of range searching are in general much harder than the standard variants, because they are not *decomposable*. For example, partitioning the query box into two sub-ranges and counting the number of colors in each sub-range tells us practically nothing about the number of colors in the full range. Halfspace colored range searching (with halfspaces as queries and points as colored objects) was studied in [15]. Orthogonal colored range searching problems (where queries are axis-parallel boxes) were studied in [4, 6, 13, 16, 19]. Additional colored range searching problems were studied in [4, 6, 14, 15]. Batched colored range intersection problems, where we report all pairs of colors (c_1, c_2) such that an object of color c_1 intersects

*Work by Micha Sharir was partially supported by NSF Grant CCF-05-14079, by a grant from the U.S.-Israeli Binational Science Foundation, by grant 155/05 from the Israel Science Fund, Israeli Academy of Sciences, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University. This work is part of the second author's M.Sc. Dissertation, prepared under the supervision of the first and third authors.

†School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: haimk@post.tau.ac.il

‡School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: rubinnat@post.tau.ac.il

§School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. E-mail: michas@post.tau.ac.il

¶School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: eladv@post.tau.ac.il

an object of color c_2, were studied in [21, 5]. A detailed review of these results is given in the full version of the paper.

Our contribution. We present a new approach to orthogonal colored range counting, based on a reduction from this problem to standard orthogonal range counting, which works in any dimension. We transform P into a higher-dimensional space, where each query box corresponds to a (negatively-oriented) *orthant*. We then apply duality, transforming query orthants into points, and input points into positively-oriented orthants. For each color $1 \le c \le C$, the space of all (points representing) query orthants containing a point with color c is the union of the respective positive orthants, denoted as $Q(c)^+$.

For each color c, we show how to decompose $Q(c)^+$ into pairwise disjoint boxes. As a consequence, a query orthant contains a point of color c iff its dual point is contained in exactly one of the boxes in the decomposition of $Q(c)^+$. Our algorithm thus collects the decomposition boxes, over all colors c, and stores them in a data structure that supports efficient containment counting queries of the form: Given a query point x, count the number of boxes that contain x.

A straightforward implementation of this technique yields a solution for the colored orthogonal range counting problem in $\mathbb{R}^d$, having query time $O(\log^{2d-1} n)$, space complexity $O(n^d \log^{2d-1} n)$ and worst case (deterministic) preprocessing time $O(n^d \log^{2d-1} n)$. A simple enhancement of the technique reduces all three performance parameters by a factor of $O(\log n)$, thus matching the performance parameters of a previous algorithm of Gupta et al. [16] for $d = 2$. If our queries are orthants, we can do better, obtaining a data structure with a query time of $O(\log^{d-1} n)$ and storage and preprocessing cost $O(n^{\lfloor d/2 \rfloor} \log^{d-1} n)$. For $d \ge 4$ and even, the same enhancement trick reduces all three performance parameters by a $\log n$ factor.

In fact, the storage size and preprocessing time of our algorithm depend on the overall number of boxes in the decomposition of $Q(c)^+$, over all colors c. This can be as large as $O(n^d)$ (for general box queries) or $O(n^{\lfloor d/2 \rfloor})$ (for orthant queries), but in practice we expect it to be much smaller. To support this statement, we show that, for random point sets (drawn independently and uniformly at random from $[0, 1]^d$), the expected number of boxes in our decomposition is only $O(n \log^{d-1} n)$, which leads to algorithms with polylogarithmic query time and near-linear expected storage and preprocessing.

Time-space tradeoff. We also consider techniques for reducing the storage (and preprocessing) at the cost of increasing the query time. In $\mathbb{R}^2$ we use a technique of Gupta et al. [12] to decompose a query $[a, b] \times [c, d]$ into two 3-sided queries $[a, b] \times (-\infty, d]$ and $[a, b] \times [c, \infty)$ on secondary structures stored at the nodes of a binary search tree over the points sorted by their y-coordinate. (This leads to significant gains, because, as we show, the structure for answering 3-sided queries requires only near-linear storage and preprocessing, whereas the structure for 4-sided queries may require near-quadratic storage.) We obtain the solution for each 3-sided query as a collection of canonical sets of colors. The difficulty is that a single color may appear in both outputs of the two 3-sided queries but we need to count it only once. As we show, counting the colors that appear twice is equivalent to computing the sizes of all pairwise intersections of the output canonical subsets. We use *sparse matrix multiplication* techniques [21, 22, 7] to precompute efficiently some of these intersection sizes, and handle others on the fly when processing a query.

We obtain a solution that has query time $O(X \log^7 n)$, and storage $O\left(\left(\frac{n}{X}\right)^2 \log^6 n + n \log^4 n\right)$, for any tradeoff parameter $1 \le X \le n$. The construction time of the data structure is (the $O^*(\cdot)$ notation hides polylogarithmic factors):

$$\begin{cases} O^*\left(\frac{n^{(\omega+1)/2}}{X^{(\omega-1)/2}}\right), & X \ge n^{\frac{\omega-1}{\omega+1}}, \\ O^*\left(\frac{n^{\frac{2-\alpha\beta+2\beta}{\beta+1}}}{X^{\frac{2-\alpha\beta}{\beta+1}}}\right), & n^{\frac{\alpha/2}{\alpha/2+1}} \le X \le n^{\frac{\omega-1}{\omega+1}}, \\ O^*(\frac{n^2}{X^2}), & X \le n^{\frac{\alpha/2}{\alpha/2+1}}. \end{cases}$$

Here ω is the smallest number such that two $t \times t$ matrices can be multiplied in time $O(t^\omega)$ (the best known upper bound on ω is 2.376), $\alpha > 0.294$ is another parameter related to matrix multiplication and $\beta = \frac{\omega-2}{1-\alpha}$ (see Section 3) [10, 11]. In particular it follows from these bounds that for $m \le n$, we can answer m queries in overall time $O^*(nm^{\frac{\omega-1}{\omega+1}}) = O(nm^{0.408})$.

Interestingly, we also show a reduction of a version of sparse matrix multiplication to an offline version of colored orthogonal range counting in $\mathbb{R}^2$. This reduction implies that if we can answer m queries on a set of n points, for $m^{\frac{1+\omega}{4}} \le n$, in $o(nm^{\frac{\omega-1}{4}}) = o(n^{1.34})$ time, then we can obtain a better algorithm for sparse rectangular matrix multiplication than the best known to date. Furthermore, if we can answer n such queries in $o(n^{\frac{2.376}{2}}) = o(n^{1.188})$ time, we improve the best known bound on ω.

A simple bucketing technique allows us to trade time for space also in dimension $d > 2$. Specifically, for any threshold parameter $1 \le X \le n$, we obtain a data structure, having query time $O(X \log^d n + \log^{2d-1} n)$ and preprocessing and storage cost $O(\frac{n^d}{X^{d-1}} \log^{2d-1} n)$. In the full version of this paper we suggest two additional techniques to improve this tradeoff for $X = \Omega^*(n^{\frac{d-2}{d-1}})$.

Colored range counting and generalized matrix multiplication. Finally, we show that colored orthogonal range counting in dimension $d > 2$ is related to the following generalization of sparse matrix multiplication, which we believe to be of independent interest. One is given a 0-1 matrix A with N nonzero entries in a sparse representation, and a list O of M d-tuples of indices of rows of A. Let t be the number of columns in A. The goal is to compute for each tuple $(i_1, \ldots, i_d) \in O$ the sum $\sum_{j=1}^{t} \prod_{k=1}^{d} A_{i_k,j}$. We call this problem *$d$-dimensional output restricted sparse matrix multiplication* ($ORSMM_d$). (Note that the case $d = 2$ asks for computing M specified entries in AA^T.) We give a reduction from this problem to colored range counting in boxes in dimension d, showing that the offline version of the latter problem is at least as hard as this generalized matrix multiplication problem. Finally, we describe efficient algorithms for $ORSMM_d$.

For lack of space, many details are omitted in this version. They can be found in the full version [20].

2 Orthogonal Colored Range Counting

Reducing to standard orthogonal range counting. We first solve the *semi-unbounded colored range counting* problem, in which the query boxes are orthants of the form $\prod_{i=1}^{d}(-\infty, a_i]$. Then we show how to reduce the colored counting problem in general boxes to the semi-unbounded case.

We represent each query orthant $\prod_{i=1}^{d}(-\infty, a_i]$ by its apex $(a_1, \ldots, a_d) \in \mathbb{R}^d$. Fix a color c, $1 \le c \le C$, and let P_c denote the subset of points of P with color c. For a point $p \in P_c$, denote by $Q_p^+ = \prod_{i=1}^{d}[p_i, \infty) \subseteq \mathbb{R}^d$, the locus of all points that represent (closed) query orthants containing p. For any point set $A \subset \mathbb{R}^d$, define $U(A) := \bigcup_{p \in A} Q_p^+$. Define the *$c$-positive region* of color c to be $Q(c)^+ = U(P_c)$. It is the locus of all points representing queries that contain at least one point of P_c. Later in this section we establish the following theorem.

THEOREM 2.1. *For any $A \subset \mathbb{R}^d$ of size n, we can decompose $U(A)$ into $B = O(n^{\lfloor d/2 \rfloor})$ pairwise disjoint boxes. Furthermore we can generate these boxes in $O(B \log^{d-1} n)$ time.*

We remark that Theorem 2.1 can be regarded as a refinement of the result of Boissonnat et al. [3] concerning the complexity of the union of n congruent axis-parallel cubes in $\mathbb{R}^d$ (think of the orthants as very large congruent cubes). As a matter of fact, our constructive proof of the theorem follows similar footsteps to those in the proof of the bound in [3]. See also [8].

Next, for each color $1 \le c \le C$, we generate the boxes in the decomposition of $Q(c)^+$, and build a standard orthogonal range searching data structure for counting the overall number of boxes containing a query point $q \in \mathbb{R}^d$. By construction (using the fact that the boxes corresponding to any fixed color are pairwise disjoint), this number is equal to the number of distinct colors that appear in the original query orthant.

To implement the data structure, we use a d-dimensional segment tree with fractional cascading at its deepest level. Let B_c be the number of boxes in the decomposition of $Q(c)^+$. Then this structure has query time $O(\log^{d-1} n)$, and space and preprocessing time $O(\sum_{1 \le c \le C} B_c \log^{d-1} n)$. These improved bounds on the storage and preprocessing time (by a logarithmic factor, as compared with standard d-dimensional segment trees) are based on the fact, established below, that all the decomposition boxes are unbounded in the positive x_1-direction. This allows us to use a simple search tree instead of a segment tree at the bottom level of the data structure, thereby saving a logarithmic factor in both storage and preprocessing time. Fractional cascading takes care of the corresponding saving in the query time. We thus obtain the following theorem.

THEOREM 2.2. *Let P be a set of n colored points in $\mathbb{R}^d$, let B_c be the number of boxes in the decomposition of $Q(c)^+$, for each color $1 \le c \le C$, and let $B = \sum_{1 \le c \le C} B_c$. Then there exists a data structure supporting semi-unbounded (orthant) colored range counting queries in $O(\log^{d-1} n)$ time, whose storage is $O(B \log^{d-1} n)$ and which can be constructed in $O(B \log^{d-1} n)$ time. In the worst, the latter bounds are both $O(n^{\lfloor d/2 \rfloor} \log^{d-1} n)$.*

General orthogonal range counting. The general problem, in which queries are arbitrary bounded axis-parallel boxes in $\mathbb{R}^d$, can be reduced to the semi-unbounded case in $\mathbb{R}^{2d}$, as follows. Denote the x_i-coordinate of a point p by $x_i(p)$, and map each point $p = (x_1(p), \ldots, x_i(p), \ldots, x_d(p)) \in P$ to the point $(x_1(p), x_1(p), \ldots, x_i(p), x_i(p), \ldots, x_d(p), x_d(p))$ in $\mathbb{R}^{2d}$, which is given the same color as the original p. Now, answering a colored range counting query $\prod_{i=1}^{d}[a_i, b_i]$ on the original point set is equivalent to answering the query $\prod_{i=1}^{d}[a_i, \infty) \times (-\infty, b_i]$ on the transformed point set. Thus we obtain the following theorems.

THEOREM 2.3. *Let P be a set of n colored points in $\mathbb{R}^d$, let $\hat{P} \subset \mathbb{R}^{2d}$ be the transformed point set of P as defined above, let B_c be the number of boxes in the decomposition of $U(\tilde{P}_c)$, for color $1 \le c \le C$, and let $B = \sum_{1 \le c \le C} B_c$. Then there exists a data* *structure supporting colored range counting queries in $O(\log^{2d-1} n)$ time, whose storage is $O(B \log^{2d-1} n)$ and which can be constructed in $O(B \log^{2d-1} n)$ time. In the worst case, the two latter bounds are $O(n^d \log^{2d-1} n)$.*

We can generalize this approach to colored orthogonal range counting queries for boxes with bounded projections on k specific coordinates, and semi-unbounded projections on the remaining $d-k$ coordinates. In this case we can reduce the problem to a semi-unbounded problem in $\mathbb{R}^{d+k}$, by duplicating the k "bounded" coordinates of each point in our input set. A query then takes $O(\log^{d+k-1} n)$ time, and the storage and preprocessing cost are both $O(n^{\lfloor (d+k)/2 \rfloor} \log^{k+d-1} n)$. As a special case, consider the colored orthogonal range counting problem on n points in the plane, where the queries are 3-sided boxes of the form $[a, b] \times (-\infty, c]$. Here $d = 2$, $k = 1$, and we obtain an algorithm with $O(\log^2 n)$ query time and space and preprocessing cost $O(n \log^2 n)$. These performance parameters are the same as those obtained by Gupta et al. [16], using a different approach based on persistence (which does not seem to extend to higher dimensions). In the full version, we apply the paradigm of [16] to extend this solution to the general case of bounded box queries, which results in a saving of a $\log n$ factor in all three performance bounds, leading to the following result.

THEOREM 2.4. *There exists a data structure for colored orthogonal range counting queries in dimension $d \geq 2$, which answers a query in $O(\log^{2d-2} n)$ time, requires $O(n^d \log^{2d-2} n)$ space, and can be constructed in $O(n^d \log^{2d-2} n)$ time.*

Decomposing the union of orthants into disjoint boxes. We next present the proof of Theorem 2.1. Let A be a set of n points in $\mathbb{R}^d$ in general position, meaning that no two points have the same x_i-coordinate, for any $i = 1, \ldots, d$.

An open *maximal empty orthant* O (with respect to A) is a negative orthant (possibly unbounded in both directions of certain coordinates), which does not contain any point of A, and is maximal with this property under inclusion. It follows that each facet of O must contain a distinct point of A in its *relative interior*. Let s_i be the point in the relative interior of the facet of O orthogonal to the x_i-axis, for $1 \leq i \leq d$. Thus $O = \prod_{i=1}^{d} (-\infty, x_i(s_i))$ (see Figure 1). (For any i for which O is unbounded in the positive x_i-direction, we take s_i to be the point at infinity whose x_i-coordinate is $+\infty$ and all other coordinates are $-\infty$.) We say that O is defined by the tuple of points $\langle s_1, \ldots, s_d \rangle$.

Our decomposition of $U(A)$ is constructed so that there is a bijection between its boxes and the maximal

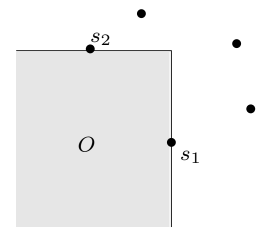

Figure 1: A maximal empty orthant

empty orthants defined by tuples $\langle s_1, \ldots, s_d \rangle$ such that $x_1(s_1) < \infty$. Specifically, let O be such a maximal empty orthant defined by $\langle s_1, \ldots, s_d \rangle$. The box in our decomposition corresponding to O is (see Figure 2 for an illustration)

$$B(O) = [x_1(s_1), \infty) \times \prod_{i=2}^{d} \left[\max_{j<i} \{x_i(s_j)\}, x_i(s_i) \right).$$

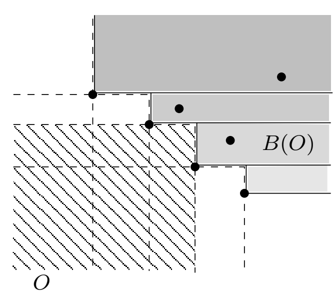

Figure 2: The box decomposition of $U(A)$.

LEMMA 2.1. *Let A be a finite point set in $\mathbb{R}^d$, and let $\mathcal{B}$ be the collection of boxes $B(O)$, where O ranges over all maximal empty orthants with respect to A, which are defined by tuples $\langle s_1, \ldots, s_d \rangle$ satisfying $x_1(s_1) < \infty$. Then the boxes of $\mathcal{B}$ are pairwise disjoint, and their union is $U(A)$.*

Proof. Follows by induction on the dimension, and omitted in this version. □

The number of boxes. Since the number of cells in the decomposition of $U(A)$ is bounded by the number of maximal empty orthants with respect to A, it suffices to bound the latter quantity.

Here is a straightforward constructive derivation of the bound $O(n^{\lfloor d/2 \rfloor})$ for this latter quantity, which resembles the analysis of Boissonnat et al. [3]. Given any empty orthant O, with some points of A on its boundary (possibly on lower-dimensional boundary faces), a *shift* of O is the operation of shrinking O by translating a facet of O in the negative direction (of the orthogonal coordinate axis). By the general position assumption, when a shift starts, exactly one point leaves the boundary of O. The shift is *legal* if such a point lies in the relative interior of a facet. A legal shift terminates as soon as one of the points of A on ∂O reaches the relative boundary of the face it is currently on, so that it now lies on a lower-dimensional face. Note that the orthant reached at the end of a legal shift is contained in the original orthant and is thus also empty.

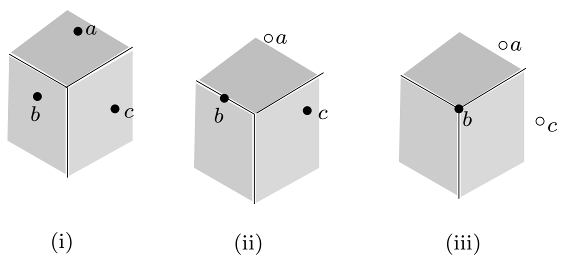

Figure 3: Two legal shifts that turn a maximal empty orthant into the one that has point b as an apex.

We start with a maximal empty orthant O, and apply to it any sequence of legal shifts, until we reach an orthant O' to which no further legal shifts can be applied. See Figure 3. Upon termination, no point of A lies in the relative interior of any facet of O'. Hence each boundary point of O' determines at least two of the coordinate values that define facets of O', and thus the number of points on $\partial O'$ is at most $\lfloor d/2 \rfloor$, which implies that the number of such empty orthants O', which can be obtained from some maximal empty orthant in the manner described above, is $O(n^{\lfloor d/2 \rfloor})$. It is also easily seen that each terminal orthant can be obtained by only $O(1)$ sequences of legal shifts, so this is also a bound on the number of maximal empty orthants, and thus also on the complexity of $U(A)$. This completes the proof of the first part of Theorem 2.1.

Output sensitive construction. The above proof can be easily converted into an algorithm that computes all maximal empty orthants in $O(n^{\lfloor d/2 \rfloor} \log^{d-1}(n))$ time. However, it is not output sensitive. In the full version we present a different algorithm, which is based on hyperplane sweeping, for constructing the B maximal empty orthants in time $O(B \log^{d-1} n)$.

We also show that the number of boxes in the decomposition of $U(A)$ is $O(B)$. This implies the second part of Theorem 2.1, and thus completes its proof. □

Random sets of points. The decomposition-based data structure of Theorem 2.3 is especially efficient, when the numbers of maximal empty orthants with respect to each of the sets $P_c \subset \mathbb{R}^d$, for $1 \le c \le C$, are all small. An open *maximal empty box* with respect to a point set P is an orthogonal axis-parallel box, which does not contain any point of P, and which is maximal with this property under inclusion (so, as in the case of orthants, each facet of the box contains a point of P in its relative interior). We allow a maximal empty box to be unbounded in certain directions, so every maximal empty orthant with respect to P is also a maximal empty box. We show that the expected number of maximal empty boxes for a random point-set with n points in $\mathbb{R}^d$ is only $O(n \log^{d-1} n)$. It would be interesting to explore the connection between this result and the known bound of $O(\log^{d-1} n)$ on the number of *maximal* points in a set of n random points in $\mathbb{R}^d$; see [1, 17].

We assume that the set P is constructed so that each point is chosen independently and uniformly at random from the uniform distribution on $[0,1]^d$. Thus, with probability 1, the points of P are in general position, and, for each i, the x_i-coordinates of the sampled points form a random permutation, which are independent of each other. We define a *t-box* to be an axis-parallel box B, that may be unbounded in certain directions, containing t points on its boundary, such that each one of the finite facets of B contains a point of P (possibly on its relative boundary). Let $B_{t,k}(P)$, for $1 \le t \le 2d$, denote the set of t-boxes containing k points of P in their interior, and put $N_{t,k}(P) = |B_{t,k}(P)|$. We denote by $B_{t,\le k}(P)$ the set of t-boxes with at most k points of P in their interior. Finally, we let $N^{(d)}_{t,k}(n)$ (resp., $N^{(d)}_{t,\le k}(n)$) denote the expected value of $|B_{t,k}(P)|$ (resp., $|B_{t,\le k}(P)|$), over the random choice of a set P of n points in $\mathbb{R}^d$. In order to bound the expected number of maximal empty boxes with respect to P, it suffices to bound $N^{(d)}_{t,0}(n)$, for all $1 \le t \le 2d$.

We prove the bound by induction on d. The case $d = 1$ is trivial: We have $N^{(1)}_{1,0}(n) = 2$, and $N^{(1)}_{2,0}(n) = n - 1$.

Assume now that $d > 1$, and that the bound holds for $d-1$. Assume also that we draw each point $p \in P$ by first drawing a point $p' \in [0,1]^{d-1}$, thereby fixing the projection of p on the hyperplane $x_d = 0$ to be p', and then drawing x_d uniformly from $[0,1]$ to be the x_d-coordinate of p.

Let $P' = \{p' \mid p \in P\}$ be the set of projections of all points in P on the hyperplane $x_d = 0$. Let $\tilde{B}_{t,0}(P)$ denote the set of empty t-boxes of P whose both facets orthogonal to the x_d-axis contain a point of P on their relative boundary. Set $\tilde{N}_{t,0}(P) = |\tilde{B}_{t,0}(P)|$ and $\tilde{N}^{(d)}_{t,0}(n) = \max_{|P|=n} \tilde{N}_{t,0}(P)$. Every t-box B in

$\tilde{B}_{t,0}(P)$ corresponds to the t-box B' of P', obtained by projecting B onto $x_d = 0$; note that B' is not necessarily empty.

Fix P', and consider the random drawings of the x_d-coordinates of the points of P. What is the probability that a t-box B' in $B_{t,k}(P')$ corresponds to an empty t-box B in $\tilde{B}^d_{t,0}(P)$? For this to happen, it is necessary and sufficient that the t boundary points of B' be consecutive in the permutation defined by the $t+k$ boundary and interior points of B' along the x_d-axis (see Figure 4). This happens with probability $\frac{t!(k+1)!}{(k+t)!} = \frac{t!}{(k+2)(k+3)\cdots(k+t)}$. Hence,

$$\mathbb{E}(\tilde{N}_{t,0}(P)) \le \sum_{k=0}^{n-t} \frac{t!}{\prod_{i=2}^{t}(k+i)} |B_{t,k}(P')|.$$

Rearranging this sum, we get:

$$\begin{aligned}\mathbb{E}(\tilde{N}_{t,0}(P)) &\le \sum_{k=0}^{n-t} \frac{t!(t-1)}{\prod_{i=2}^{t+1}(k+i)} |B_{t,\le k}(P')| \\ &+ \frac{t!}{\prod_{i=2}^{t}(n-t+i)} |B_{t,\le n-t}(P')|.\end{aligned}$$

This holds for every choice of P', so if we average over the choices of P', we obtain the following recurrence:

$$\begin{aligned}\tilde{N}^{(d)}_{t,0}(n) &\le \sum_{k=0}^{n-t} \frac{t!(t-1)}{\prod_{i=2}^{t+1}(k+i)} N^{(d-1)}_{t,\le k}(n) \\ &+ \frac{t!}{\prod_{i=2}^{t}(n-t+i)} N^{(d-1)}_{t,\le n-t}(n).\end{aligned} \tag{2.1}$$

We next consider empty t-boxes, whose both facets orthogonal to the x_d-axis do not contain points of P in their relative boundaries. We only consider boxes that are not strips bounded by one or two hyperplanes orthogonal to the x_d-direction (there are $O(n)$ such strips). Any such box B, can be charged to a box in $\tilde{B}_{t',0}(P)$, for some $t-2 \le t' \le t$, by applying two legal shifts to B, the first shifting the top facet down, and the second shifting the bottom facet up. With some care, this also applies to boxes that are unbounded in the x_d-direction. The resulting box $\tilde{B}$ is uniquely charged in this manner. Similarly, an empty t-box B, such that one of its facets orthogonal to the x_d-axis does not contain a point in the relative boundary, can be charged to a box $\tilde{B}$ in $\tilde{B}_{t',0}(P)$, for some $t-1 \le t' \le t$. This can be done by applying just one legal shift that shifts that facet. Such a $\tilde{B}$ is charged at most twice.

Hence, denoting by $\tilde{N}^d_{t,0}(n)$ the expected value of $\tilde{N}_{t,0}(P)$, over the random choice of P, we get

$$N^{(d)}_{t,0}(n) = O(\tilde{N}^{(d)}_{t,0}(n) + \tilde{N}^{(d)}_{t-1,0}(n) + \tilde{N}^{(d)}_{t-2,0}(n) + n). \tag{2.2}$$

To estimate $N^{(d-1)}_{t,\le k}(n)$, note that if we sample a random subset of P of size n/k, we obtain a random set of n/k points drawn independently from the same uniform distribution as P. Hence, we can apply the Clarkson-Shor probabilistic technique [9] to conclude that

$$N^{(d-1)}_{t,\le k}(n) = O(k^t N^{(d-1)}_{t,0}(n/k)). \tag{2.3}$$

By the induction hypothesis we have $N^{(d-1)}_{t,0}(n) = O(n \log^{d-2} n)$, for any $1 \le t \le 2d-2$, and for any n. Hence, by (2.3), $N^{(d-1)}_{t,\le k}(n) = O(nk^{t-1} \log^{d-2} n)$, for any $1 \le t \le 2d-2$. Plugging this into Inequality (2.1) and using Inequality (2.2), we obtain the bound $N^{(d)}_{t,0}(n) = O(n \log^{d-1} n)$, for $1 \le t \le 2d$.

We thus have proven the following theorem.

Theorem 2.5. *Let P be a set of n points in $\mathbb{R}^d$, drawn independently from the uniform distribution on $[0,1]^d$. Then the expected number of maximal empty boxes with respect to P is $O(n \log^{d-1} n)$, where the constant of proportionality depends on d.*

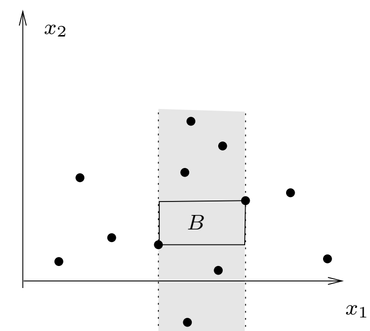

Figure 4: Bounding the expected number of empty boxes in the plane. The highlighted box B belongs to $\tilde{B}_{2,0}(P)$, and its x-projection belongs to $B_{2,5}(P')$.

In particular, the bound of Theorem 2.5 also applies to the expected number of maximal empty orthants. Using Theorems 2.2 and 2.3, and omitting some details related to the extension to the case of bounded box queries, we obtain the following result.

Theorem 2.6. *Let P be a random colored point-set of cardinality n in $\mathbb{R}^d$, such that the points of each color are drawn as above. Then there exists a data structure supporting semi-unbounded (resp., bounded) colored range counting queries in $O(\log^{d-1} n)$ (resp., $O(\log^{2d-1} n)$) time, whose expected storage and preprocessing costs are both $O(n \log^{2d-2} n)$ (resp., $O(n \log^{3d-2} n)$).*

Remark: The enhancement that shaves off a logarithmic factor, as provided in Theorem 2.4, does not apply in this case, because it is based on n copies of the structure, and the expected number of maximal empty orthants may be linear in each subproblem, giving rise to an overall data structure of super-quadratic size.

3 Time-Space Tradeoff

In this section we present several techniques for reducing storage at the expense of increasing query time.

The planar case: Splitting boxes. Our tradeoff technique for the planar case uses the more space efficient data structure of Theorem 2.2 that supports colored range counting queries with 3-sided boxes of the form $[a, b] \times (-\infty, d]$ or $[a, b] \times [c, \infty)$. Recall that this problem can be transformed into a standard orthogonal containment searching in $\mathbb{R}^3$, and that the resulting data structure requires only $O(n \log^2 n)$ storage and preprocessing cost. For the present solution, instead of just counting colors, we want to output the set of all colors in the query 3-sided box as a union of canonical subsets of colors. Since the data structure has three levels, the number of such sets in a query output is $O(\log^3 n)$, and each color appears in at most one of these sets (because of the disjointness of the boxes in the decomposition of any single $U(P_c)$).

The full data structure. We store the points of P, in the increasing order of their y-coordinates, at the leaves of a balanced binary tree T. At each internal node v, we store a pair of auxiliary data structures, one for answering queries of the form $[a, b] \times [c, \infty)$, and one for queries of the form $[a, b] \times (-\infty, d]$. The first (resp., second) structure is built on the points stored at the left (resp., right) subtree of v. We also store at v a y-coordinate $Y(v)$ that separates the y-coordinates of the points stored at the left subtree from those stored at the right subtree of v. If v is a leaf, we let $Y(v)$ be the y-coordinate of the singleton point stored at v.

For each node t at the deepest (third) level of some auxiliary structure, denote by $c(t)$ the set of colors represented in the canonical subset of t.

Let $q = [a, b] \times [c, d]$ be a query box. We search down the tree T with $[c, d]$, and denote by v_q the highest node for which $[c, d]$ contains $Y(v_q)$. If v_q is a leaf, we check whether the single point that it stores lies in q, and return 1 if its does and 0 otherwise. If v_q is an internal node, we query the two structures at v_q with $[a, b] \times [c, \infty)$ and with $[a, b] \times (-\infty, d]$, respectively. Let D_q (resp., U_q) denote the set of $O(\log^3 n)$ nodes corresponding to the canonical subsets that are returned by the former (resp., latter) sub-query. Observe that the set of colors in q is exactly the union of the canonical sets of colors associated with the nodes in $U_q \cup D_q$. (See [12] for more details.) Hence, using the exclusion-inclusion principle, the number of colors that appear in q is equal to

$$\sum_{s \in U_q} |c(s)| + \sum_{t \in D_q} |c(t)| - \sum_{s \in U_q,\ t \in D_q} |c(s) \cap c(t)|. \tag{3.4}$$

Ideally, we would like to have for every $s \in U_q$ and $t \in D_q$ the value of $|c(s) \cap c(t)|$ pre-stored. However, doing this for every possible pair of canonical sets would be too expensive, and may result in super-quadratic space complexity. Nevertheless, if we did have these values available, answering a query could then be done in $O(\log^6 n)$ time, using (3.4).

Instead, we derive a tradeoff between storage and query time, determined by a threshold parameter X in the range $1 \le X \le n$. Let $\mathcal{D}(v)$ (resp., $\mathcal{U}(v)$) be the set of canonical nodes in the auxiliary structure of node $v \in T$ used for answering queries of type $[a, b] \times [c, \infty)$ (resp., $[a, b] \times (-\infty, d]$). A node $t \in \mathcal{D}(v) \cup \mathcal{U}(v)$ is called X*-heavy* if $|c(t)| > X$; otherwise it is called X*-light*. For every node $v \in T$ we construct and store, as part of the preprocessing stage, a matrix $M(v)$, whose rows and columns correspond to the X-heavy nodes in $\mathcal{D}(v)$ and $\mathcal{U}(v)$, respectively. For each pair of X-heavy canonical nodes $s \in \mathcal{U}(v)$ and $t \in \mathcal{D}(v)$, we store in $M_{s,t}(v)$ the value of $|c(t) \cap c(s)|$. Let n_v be the number of points stored at the subtree of T rooted at v. The overall size of all the canonical subsets associated with the nodes of $\mathcal{U}(v) \cup \mathcal{D}(v)$ is $O(n_v \log^3 n)$. Hence, the number of X-heavy canonical nodes in $\mathcal{D}(v) \cup \mathcal{U}(v)$ is $O\left(\frac{n_v}{X} \log^3 n\right)$, so $M(v)$ has size $O\left(\left(\frac{n_v}{X}\right)^2 \log^6 n\right)$. (The time needed to construct $M(v)$, for all nodes $v \in T$, is discussed later.) Summing this bound over all nodes v of T, we get an overall bound of $O(\left(\frac{n}{X}\right)^2 \log^6 n)$.

For each canonical node $t \in \mathcal{D}(v) \cup \mathcal{U}(v)$, we also store a dictionary data structure on $c(t)$ (implemented as a binary search tree), supporting logarithmic-time searches. Clearly, since $\sum_{t \in \mathcal{D}(v) \cup \mathcal{U}(v)} |c(t)| = O(n_v \log^3 n)$, these additional structures take total of $O(n \log^4 n)$ extra storage.

The total space complexity of our solution is thus $O\left(\left(\frac{n}{X}\right)^2 \log^6 n + n \log^4 n\right)$.

Handling queries. It suffices to describe how to compute $|c(s) \cap c(t)|$ efficiently, for each $s \in U_q$ and $t \in D_q$. If both s and t are X-heavy, then this value is stored in $M(v_q)$ and we simply retrieve it. Otherwise, we can assume, without loss of generality, that $|c(s)| \le X$. In this case we check, for each $c \in c(s)$, whether $c \in c(t)$, and count the number of such colors. This takes a total of $O(X \log n)$ time. We repeat this for each pair $(s, t) \in U_q \times D_q$, for a total of $O(X \log^7 n)$ query time.

Preprocessing. Clearly, T with all its auxiliary data structures can be constructed in time $O(n\log^3 n)$ and the dictionary structures at the nodes of $|\mathcal{U}\cup\mathcal{D}|$ can be constructed in overall time $O(n\log^4 n)$. It remains to describe the construction of matrices M_v.

Denote by $\mathcal{D}$ (resp., $\mathcal{U}$) the set of canonical nodes in all the auxiliary structures used for answering queries of type $[a,b]\times[c,\infty)$ (resp., $[a,b]\times(-\infty,d]$). Let M_D (resp., M_U) denote the matrix whose rows correspond to the X-heavy nodes of $\mathcal{D}$ (resp., of $\mathcal{U}$), and whose columns correspond to the colors $1,\ldots,C$, such that, for $t\in\mathcal{D}$ (resp., $t\in\mathcal{U}$) and color c, the (t,c)-entry of the matrix is 1 if $c\in c(t)$ and 0 otherwise. It follows that $M = M_D {M_U}^T$ contains all the matrices $M(v)$ as submatrices.

Using that $\sum_v n_v = O(n\log n)$, we get that each of the matrices M_D and M_U has $t = O\left(\frac{n}{x}\log^4 n\right)$ rows and $N = O(n\log^4 n)$ non-zero entries. Hence, we can construct M using the sparse rectangular matrix multiplication technique of Kaplan *et al.* [21] (which extends a technique of Yuster and Zwick [22] and of Chan [7]). Specifically, for two matrices A, B, each having t rows and $\le N$ non-zero items, these methods construct AB^T in time

$$\begin{cases} O(Nt^{\frac{\omega-1}{2}}) & \text{if } N\ge t^{\frac{\omega+1}{2}},\\ O(N^{\frac{2\beta}{\beta+1}}t^{\frac{2-\alpha\beta}{\beta+1}}) & \text{if } t^{1+\frac{\alpha}{2}}\le N\le t^{\frac{\omega+1}{2}},\\ O(t^2) & \text{if } N\le t^{1+\frac{\alpha}{2}}.\end{cases}$$

Here (i) $\omega < 2.376$ is the exponent of matrix multiplication, i.e., ω is the smallest number such that two $t\times t$ matrices can be multiplied in time $O(t^\omega)$; (ii) $\alpha > 0.294$ is the largest value of r for which a $t\times t^r$ matrix and a $t^r\times t$ matrix can be multiplied in time $O(t^2)$; and (iii) $\beta = \frac{\omega-2}{1-\alpha}\approx 0.533$, see [18]. Hence, the construction time of the data structure is:

$$\begin{cases} O^*\left(\frac{n^{(\omega+1)/2}}{X^{(\omega-1)/2}}\right) & X\ge x_1(n),\\ O^*\left(\frac{n^{\frac{2-\alpha\beta+2\beta}{\beta+1}}}{X^{\frac{2-\alpha\beta}{\beta+1}}}\right) & x_1(n)\ge X\ge x_2(n),\\ O^*(\frac{n^2}{X^2}) & x_2(n)\ge X,\end{cases}$$

where $x_1(n) = n^{\frac{\omega-1}{\omega+1}}\approx n^{0.408}$ and $x_2(n) = n^{\frac{\alpha/2}{\alpha/2+1}}\approx n^{0.128}$.

Optimizing for a fixed number of queries. Suppose next that we know (or guess) in advance the number m of queries. We can then optimize the choice of X, so as to minimize the overall time for answering m colored range counting queries, including the time spent at the preprocessing stage. A simple calculation yields:

(a) $m\le n$ queries can be answered in overall time $O^*(nm^{\frac{\omega-1}{\omega+1}}) = O(nm^{0.408})$. (b) $n\le m\le n^{1.616}$ queries can be answered in overall time $O(n^{0.862}m^{0.546})$. (c) $m\ge n^{1.616}$ queries can be answered in overall time $O^*(m^{2/3}n^{2/3})$.

In particular, n colored range counting queries can be answered in overall time $O(n^{1.408})$, including time spent at the preprocessing stage.

Tradeoff in higher dimensions. In higher dimensions, there are other approaches to achieving a tradeoff between query time and storage. For lack of space, we describe only one of them, and refer to the full paper [20] for more details.

Bucketing. Partition the set of colors into $O(\log n)$ "buckets" $\mathcal{C}_i$, such that $c\in\mathcal{C}_i$ if and only if $2^{i-1}\le |P_c| < 2^i$. Put $C_i := |\mathcal{C}_i|$, and let n_i be the number of points having colors in $\mathcal{C}_i$. Clearly, $n_i = \Theta(2^i C_i)$. We solve the colored range counting problem separately within each $\mathcal{C}_i$, and output the sum of the counts obtained in each of these $O(\log n)$ subproblems.

Again, we fix some threshold parameter $1\le X\le n$. Answering a colored range counting query with a box q within $\mathcal{C}_i$ depends on the relationship between X and C_i. If $X\ge C_i$, we simply test, for each color c in $\mathcal{C}_i$, whether $q\cap P_c\ne\emptyset$, and count up the number of colors with this property. Using the standard orthogonal range searching machinery [2], this takes $O(\log^{d-1}|P_c|) = O(i^{d-1})$ time per color c, for a total of $O(i^{d-1}C_i) = O(i^{d-1}X)$ time. Summing these bounds over all buckets with $X\ge C_i$, we obtain a total of $O(X\log^d n)$ time.

If $X < C_i$, we use the technique of Theorem 2.3 for answering the query. The query time is $O(\log^{2d-1} n_i)$, and the space and preprocessing cost are both

$$O\left(C_i\cdot 2^{id}\log^{2d-1} n_i\right) = O\left(\frac{n_i^d}{X^{d-1}}\log^{2d-1} n\right)$$

(we use here the facts that $n_i/C_i = \Theta(2^i)$ and that $X < C_i$). Hence, summing over the buckets, and adding the costs for buckets with $X\ge C_i$, the overall query time is $O(X\log^d n + \log^{2d-1} n)$. (For this, we use a single colored range counting data structure for all the buckets with $C_i > X$.) The overall preprocessing and space complexity is $O\left(\frac{n^d}{X^{d-1}}\log^{2d-1} n\right)$. Optimizing the choice of X, we can answer m colored range counting queries in time $O(nm^{1-1/d}\log^{d+1-1/d} n + m\log^{2d-1} n)$, including time spent at the preprocessing stage.

4 Hardness of Orthogonal Colored Range Counting

Consider the following *Output Restricted Sparse Matrix Multiplication* problem (*ORSMM*). The input is a sparse matrix A with N non-zero entries (and therefore at most N rows and columns), and a set O of M pairs, (i,j), where i and j are indices of two rows of A. The goal is to compute, for each pair $(i, j) \in O$, the (i, j)-th entry of the product AA^T. We further assume here that A is boolean.

The offline version of the colored orthogonal range counting problem with n points and m queries is closely related to the $ORSMM$ problem, as asserted in the following result.

Theorem 4.1. *(a) The 2-dimensional orthogonal colored range counting problem on n points and m query rectangles can be reduced to an $ORSMM$ problem, where the matrix A has $N = O(n \log^4 n)$ non-zero entries and we ask for $M = O(m \log^6 n)$ entries of the AA^T. The reduction takes $O(n \log^4 n)$ time.*

(b) Conversely, the $ORSMM$ problem, for a Boolean matrix A with N non-zero entries, where we need to compute M output pairs, can be reduced in linear time to a 2-dimensional colored orthogonal range counting problem on $O(N)$ points and M query rectangles.

Proof. Part (a) follows from the data structure is Section 3. We prove part (b). We can restate the $ORSMM$ problem as follows. Let X be the set of columns in the matrix A. For each row i, let $S_i \subseteq X$ denote the the set of columns where row i has ones. Let O be the set of M output pairs to be computed. For each pair $(i, j) \in O$ we have to compute $|S_i \cap S_j|$, which is the (i, j)-th entry of AA^T. Since $|S_i \cap S_j| = |S_i| + |S_j| - |S_i \cup S_j|$, this is equivalent to computing, for each pair $(i, j) \in O$, the quantity $|S_i \cup S_j|$.

Let k denote the number of nonempty rows of A. We construct a colored range counting instance where the points are $p_1 = (1, 1)$, $p_2 = (2, 2)$, ..., $p_k = (k, k)$ and $p'_1 = (k + 1, 1)$, $p'_2 = (k + 2, 2)$, ..., $p'_k = (2k, k)$. We assign a distinct color to each column of A. We next replace each point p_i by $|S_i|$ points, which are close to each other within distance $\epsilon \ll 1$ of p_i. We color these points by the colors of the columns in S_i. We do the same for each of the points p'_i. Let P denote the resulting point set; we have $|P| = 2N$. Clearly, with an appropriate representation of the input, this construction takes $O(N)$ time.

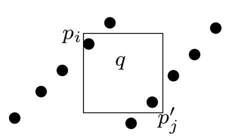

Figure 5: The rectangle q contains only p_i and p'_j.

Now, in order to calculate $|S_i \cup S_j|$, for $j \le i$, we query P with the rectangle $q = [i - \epsilon, k + j + \epsilon] \times [j - \epsilon, i + \epsilon]$. Since q contains only the points p_i and p'_j, it is clear that the number of distinct colors in q is $|S_i \cup S_j|$. See Figure 5. □

For any $t \times t$ matrix A, computing AA^T is a special case of $ORSMM$ where $N = M = t^2$. The best known algorithm for computing AA^T runs in $O(t^\omega) = O(N^{\omega/2})$ time, for $\omega \simeq 2.376$. So any algorithm for $ORSMM$ whose running time is faster than $O(\min(N, M)^{\omega/2})$ would immediately imply a better algorithm for boolean matrix multiplication, thereby solving a long-standing open problem. Using Theorem 4.1, we obtain that an algorithm for planar colored orthogonal range counting that can answer m box queries with respect to a set of $O(n)$ colored points in $o(\min(n, m)^{\omega/2})$ time would imply an algorithm that can compute AA^T in $o(t^\omega)$ time, for any $t \times t$ matrix A.

Similarly, consider the problem of computing AA^T where A is a sparse rectangular Boolean matrix with t rows and N ones such that $N \ge t^{\frac{\omega+1}{2}}$. The best known algorithm for performing this computation runs in $O(Nt^{\frac{\omega-1}{2}})$ time (see Section 3). For any such matrix A, computing AA^T is also an instance of $ORSMM$ with N ones and $M = t^2$ pairs. So an algorithm for $ORSMM$ that runs in $o(NM^{\frac{\omega-1}{4}})$ time, for $N \ge M^{\frac{\omega+1}{4}}$, would imply a faster algorithm for sparse rectangular matrix multiplication than the best known to date. Using Theorem 4.1, we obtain that the existence of an algorithm for planar colored orthogonal range counting that can answer m box queries with respect to a set of n colored points in $o(nm^{\frac{\omega-1}{4}})$ time, for $n \ge m^{\frac{1+\omega}{4}}$, would also imply an algorithm that can compute AA^T for any rectangular matrix A with t rows and N ones such that $N \ge t^{\frac{\omega+1}{2}}$, faster than what is known to date.

Generalized matrix multiplication and higher-dimensional colored range counting. For $d > 2$ we can show a similar relation between colored orthogonal range counting and a generalization of $ORSMM$. In this generalization, one is given a boolean matrix A with N nonzero entries in a sparse representation, and a list O of M d-tuples of indices of rows of A. Let t be the number of columns in A. The goal is to compute, for each tuple $(i_1, \ldots, i_d) \in O$, the sum $\sum_{j=1}^{t} \prod_{k=1}^{d} A_{i_k, j}$. We call this problem the *d-dimensional output restricted sparse matrix multiplication* problem, and denote it by $ORSMM_d$. The following theorem generalizes Theorem 4.1 to dimension $d > 2$.

Theorem 4.2. *Any instance of the $ORSMM_d$ problem, of a Boolean matrix A with N nonzero entries and M output tuples, can be reduced in linear time, to $O(1)$ instances of d'-dimensional colored orthogonal range counting, for $d' \le d$, each on $O(N)$ points and M*

query boxes.

Proof. A straightforward extension of the preceding proof, and omitted in this version. □

Efficient algorithms for $ORSMM_d$. We obtain an efficient algorithm for $ORSMM_d$ in three stages, where each stage uses the previous algorithm as a subroutine. Omitting all details, which can be found in the full version, we have:

THEOREM 4.3. *$ORSMM_d$ with N nonzero entries and M output tuples can be solved in time $O^*\left(NM^{\frac{(d-1)(\omega+d-3)}{d+(d-1)(\omega+d-3)}}\right)$, for $N \geq M^{\frac{\omega+2d-3}{d+(d-1)(\omega+d-3)}}$; $O^*\left(N^{\frac{(d-\alpha\beta+\beta-1)(d-1)+d+\beta-1}{(d-\alpha\beta)(d-1)+d+\beta-1}} M^{1-\frac{d+\beta-1}{(d-1)(d-\alpha\beta)+d+\beta-1}}\right)$, for $M^{\frac{1+\alpha-\alpha/d}{d-\alpha+\alpha/d}} \leq N \leq M^{\frac{d+\omega+d-3}{d+(d-1)(\omega+d-3)}}$, and $O^*\left(N^{\frac{d}{d+1}} M^{\frac{d}{d+1}}\right)$, for $N \leq M^{\frac{1+\alpha-\alpha/d}{d-\alpha+\alpha/d}}$.*

For $d = 2$, we get a solution for $ORSMM_2$ that takes $O^*\left(NM^{\frac{\omega-1}{\omega+1}}\right)$ time, for $M \leq N$. This is inferior to the one given by [21] for the case $M = \binom{t}{2}$ (no output restriction). It would be interesting to improve the new algorithm for all possible values of M.

References

[1] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson, On the average number of maxima in a set of vectors and applications, J. ACM 25 (1978), 536–543.

[2] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd Edition, Springer Verlag, Heidelberg, 2000.

[3] J.-D. Boissonnat, M. Sharir, B. Tagansky and M. Yvinec, Voronoi diagrams in higher dimensions under certain polyhedral distance functions, *Discrete Comput. Geom.* 19 (1998), 485–519.

[4] P. Bozanis, N. Kitsios, C. Makris, and A. Tsakalidis, New upper bounds for generalized intersection searching problems, *Proc. 22nd International Colloquium on Automata, Languages and Programming*, volume 944 of Lecture Notes in Computer Science, pages 464–475, Berlin,1995, Springer-Verlag.

[5] P. Bozanis, N. Kitsios, C. Makris, and A. Tsakalidis, New results on intersection query problems, *The Computer Journal* 40 (1997), 22–29.

[6] P. Bozanis, N. Kitsios, C. Makris, and A. Tsakalidis, Red-blue intersection reporting for objects of non-constant size, *The Computer Journal* 39 (1996), 541–546.

[7] T. Chan, Dynamic subgraph connectivity with geometric applications, Proc. 34th ACM Sympos. Theory Comput. (2002), 7–13.

[8] L. P. Chew, D. Dor, A. Efrat, and K. Kedem, Geometric pattern matching in d-dimensional space, Discrete and Computational Geometry 21 (1999), 257–274.

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

[10] D. Coppersmith. Rectangular matrix multiplication revisited, *J. Complexity* 13(1997):42–49.

[11] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions, *J. Symb. Comput.* 9(1990),251–280.

[12] D. Gupta, R. Janardan and M. Smid, Computational geometry: Generalized intersection searching, Chapter 64 in *Handbook of Data Structures and Applications*, D. Mehta and S. Sahni (editors), Chapman & Hall/CRC, Boca Raton, FL, 64-1–64-17, 2005.

[13] D. Gupta, R. Janardan and M. Smid, A technique for adding range restrictions to generalized searching problems, *Information Procesing Letters* 69 (1999), 7–13.

[14] D. Gupta, R. Janardan and M. Smid, Algorithms for some intersection searching problems involving circular objects , *International Journal of Mathematical Algorithms* 1 (1999), 35–52.

[15] D. Gupta, R. Janardan and M. Smid, Algorithms for generalized halfspace range searching and other intersection searching problems, *Computational Geometry: Theory and Applications*, 5 (1996), 321–340.

[16] D. Gupta, R. Janardan and M. Smid, Further results on generalized intersection problems: Counting, Reporting, and Dynamization, *Journal of Algorithms* 19 (1995), 282–317.

[17] S. Har-Peled, On the expected complexity of random convex hulls. Technical Report 330/98, School Math. Sci., Tel-Aviv Univ., Tel-Aviv, Israel, 1998.

[18] X. Huang and V.Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complexity* 14 (1998), 257–299.

[19] R. Janardan and M. Lopez. Generalized intersection searching problems, *International Journal of Computational Geometry and Applications*, 3 (1993), 39–69.

[20] H. Kaplan, N. Rubin, M. Sharir and E. Verbin, Counting colors in boxes, `http://www.cs.tau.ac.il/~rubinnat/clb.pdf`.

[21] H. Kaplan, M. Sharir and E. Verbin, Colored intersection searching via sparse rectangular matrix multiplication, *Proc. 22th ACM Symp. on Computational Geometry* (2006), 52–60.

[22] R. Yuster and U. Zwick, Fast sparse matrix multiplication, *ACM Trans. Algorithms* 1 (2005), 2–13.