Low Space Data Structures for Geometric Range Mode Query

Stephane Durocher* Hicham El-Zein† J. Ian Munro† Sharma V. Thankachan†

Abstract

Let $\mathcal{S}$ be a set of n points in an $[n]^d$ grid, such that each point is assigned a color. Given a query range $\mathcal{Q} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$, the geometric range mode query problem asks to report the most frequent color (i.e., a mode) of the multiset of colors corresponding to points in $\mathcal{S} \cap \mathcal{Q}$. When $d = 1$, Chan et al. [2] gave a data structure that requires $O(n + (n/\Delta)^2/w)$ words of space and supports range mode queries in $O(\Delta)$ time for any $\Delta \geq 1$, where $w = \Omega(\log n)$ is the word size. Chan et al. also proposed a data structures for higher dimensions (i.e., $d \geq 2$) with $O(s_n + (n/\Delta)^{2d})$ space and $O(\Delta \cdot t_n)$ query time, where s_n and t_n denote the space and query time of a data structure that supports orthogonal range counting queries on the set $\mathcal{S}$. In this paper we show that the space can be improved without any increase to the query time, by presenting an $O(s_n + (n/\Delta)^{2d}/w)$-space data structure that supports orthogonal range mode queries on a set of n points in d dimensions in $O(\Delta \cdot t_n)$ time, for any $\Delta \geq 1$. When $d = 1$, these space and query time costs match those achieved by the current best known one-dimensional data structure.

1 Introduction

Range query problems have proven to be of fundamental importance in computational geometry, both as tools employed to provide efficient solutions to various geometric problems, and also in the study of their optimality with respect to space and query time. In this paper we investigate the range mode query problem in a multi-dimensional setting:

Definition 1 (Range Mode Query) *Given $\mathcal{S}$, a set of n points in an $[n]^d$ grid, such that each point is a assigned a color (there can be at most n colors). A* range mode query $\mathcal{Q} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$ *asks for the most frequent color in $\mathcal{S} \cap \mathcal{Q}$.*

Although the one-dimensional range query problem has received significant attention [3, 8, 10, 9, 6], only limited attention has been paid to the multi-dimensional problem. The first solution for the multi-dimensional case was proposed recently by Chan et al. [3]. They gave a data structure that requires $O(s_n + (n/\Delta)^{2d})$ words of space and supports d-dimensional range mode queries in $O(\Delta \cdot t_n)$ time for any $\Delta \geq 1$, where s_n is the space of an orthogonal range counting data structure in d dimensions with query time t_n. The model of computation is the standard Word RAM model with word size $w = \Omega(\log n)$. In this paper we show that the space of the range mode query data structure can be improved to $O(s_n + (n/\Delta)^{2d}/w)$ words while maintaining the same query time. That is, our data structure achieves the same asymptotic space and query time costs as those of the current best known range mode query data structure for one-dimensional data [2].

1.1 Related Work

The first range mode data structure (on arrays) was proposed by Krizanc et al. [8], requiring $O(n)$ space for $O(\sqrt{n} \log\log n)$ query time. They also described data structures that provides constant query time using $O(n^2 \log\log n/\log n)$ space, and $O(n^\epsilon \log n)$ query time using $O(n^{2-2\epsilon})$ space. Petersen and Grabowski [10] improved the first bound to constant time and $O(n^2 \log\log n/\log^2 n)$ space. Peterson [9] later improved the second bound to $O(n^\epsilon)$ time queries using $O(n^{2-2\epsilon})$ space for any $\epsilon \in (0, 1/2]$. Chan et al. [3] further improved the last bound to $O(n^\epsilon)$ time queries using $O(n^{2-2\epsilon}/\log n)$ space. Using reductions from boolean matrix multiplication, they showed that query times significantly lower than $\sqrt{n}$ are unlikely for this problem with linear space [3]. Finally, Greve et al. [6] proved a lower bound of $\Omega(\log n/\log(s \cdot w/n))$ time for any data structure that supports range mode query on arrays using s memory cells of w bits in the cell probe model.

Given a fixed $\alpha \in (0, 1]$ and a range $\mathcal{Q}$, the objective of an approximate range mode query is to return an element whose frequency in $\mathcal{S} \cap \mathcal{Q}$ is at least $\alpha \cdot m$, where m denotes the frequency of the mode of $\mathcal{S} \cap \mathcal{Q}$. Bose et al. [1] gave a data structure that requires $O(n/(1-\alpha))$ space and answers approximate range mode queries in $O(\log\log_{1/\alpha}(n))$ time, as well as a data structure that answers queries in constant time when $\alpha \in \{1/2, 1/3, 1/4\}$, using $O(n \log(n))$, $O(n \log\log(n))$, and $O(n)$ space respectively. Greve et al. [6] improved previous results by giving a data structure that supports range mode queries in $O(1)$ time using $O(n)$

*University of Manitoba, Winnipeg, Canada, durocher@cs.umanitoba.ca

†University of Waterloo, Waterloo, Canada, {helzein,imunro,thanks}@uwaterloo.ca

space when $\alpha = 1/3$, and $O(\log(\alpha/(1-\alpha)))$ time using $O(n\alpha/(1-\alpha))$ space when $\alpha \in [1/2, 1)$.

Another related question is the problem of finding a least frequent element (with frequency at least one) in a one dimensional range. Chan et al. [4] gave the first solution with linear space and $O(\sqrt{n})$ query time. Later, Durocher et al. [5] improved the query time to $O(\sqrt{n/w})$. See the recent survey by Skala [11] for further reading.

2 Framework

A point $p \in \mathcal{S}$ is represented by a $(d+1)$-tuple $(x_1, x_2, \ldots, x_d, c)$, where for each i, $p.x_i$ is p's coordinate in dimension i, and $p.c$ is the color associated with p. When d is constant, we can map the input set $\mathcal{S}$ to rank space using standard techniques,[1] requiring $O(n)$ words of additional space and an $O(\log n)$ additive increase to query time to map any point in rank space back to its original value. Throughout the paper we assume that points are in rank space. That is for any point $p \in \mathcal{S}$ and any $i \in \{1, \ldots, d\}$, $p.x_i \in \{0, \ldots, n-1\}$. Moreover in each dimension the coordinates are unique. Thus we have the following lemma:

Lemma 1 *The number of points in a rectangle $\mathcal{Q} = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \ldots \times [\alpha_d, \beta_d]$ is equal to the minimum element in $\{\beta_i - \alpha_i + 1 \mid 1 \le i \le d\}$.*

Definition 2 *Let $\Delta \ge 1$ be an integer. A Δ-box is a region $R = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \ldots \times [\alpha_d, \beta_d]$, where for all i, $\alpha_i = k\Delta$ and $\beta_i = k'\Delta$ for some integers k and k'.*

There are $(1 + \lfloor (n-1)/\Delta \rfloor)^{2d} = \Theta((n/\Delta)^{2d})$ distinct Δ-boxes in our rank space grid, which includes empty boxes, i.e., boxes with $\alpha_i \ge \beta_i$ for some $i \in [1, d]$. Each Δ-box $R = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \ldots \times [\alpha_d, \beta_d]$ can be identified using a unique index, given by:

$$rank(R, \Delta) = \sum_{i=1}^{d} ((\alpha_i/\Delta) \cdot \phi^{2i-2} + (\beta_i/\Delta) \cdot \phi^{2i-1}),$$

where $\phi = 1 + \lfloor (n-1)/\Delta \rfloor$. Notice that $rank(R, \Delta)$ can be computed in $O(d)$ time (i.e., constant time when d is constant) given any R and Δ.

[1]For $k = 1, 2, ..., d$, let $E_k[0, n-1]$ be an array of length n sorted in ascending order such that the entries in E_k represent the kth coordinates of the points in $\mathcal{S}$. A point $p \in \mathcal{S}$ maps to the point $p'(z_1, z_2, ..., z_d, p.c)$ in rank space, where $E_k[z_k]$ is equal to the kth coordinate of p. The total space for maintaining these arrays is $d \cdot n$ words.

A query $\mathcal{Q} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$ on $\mathcal{S}$ maps to an equivalent query $\mathcal{Q}^* = [a_1^*, b_1^*] \times [a_2^*, b_2^*] \times \ldots \times [a_d^*, b_d^*]$ in rank space, where $E_k[a_k^* - 1] < a_k \le E_k[a_k^*]$ and $E_k[b_k^*] \le b_k < E_k[b_k^* + 1]$. We can obtain $\mathcal{Q}^*$ from $\mathcal{Q}$ in $O(d \log n)$ time by applying $2d$ binary search operations.

3 Data Structure of Chan et al.

In this section we describe the data structure presented by Chan et al. [3]. The data structure relies on the following observation [8]: a mode of $\mathcal{Q}_1 \cup Q_2$ is either a mode of $\mathcal{Q}_1$ or an element in $\mathcal{Q}_2$. Throughout Sections 3 and 4 we assume that d is a constant.

Data Structure. The data structure consists of two components:

1. An array A of length $(1 + n/\Delta)^{2d}$, such that for each Δ-box R and $i = rank(R, \Delta)$, $A[i]$ stores the mod of R.

2. For each color c, maintain an orthogonal range counting data structure over the set of points in $\mathcal{S}$ with color c. The total space and query time can be bounded by s_n and t_n, where s_n is the space of an orthogonal range counting data structure over n points in d dimensions and t_n is its query time.

Therefore the total space used is $O(s_n + (n/\Delta)^{2d})$ words.

Query Algorithm. To answer a query $\mathcal{Q} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$, first find the largest rectangle $\mathcal{Q}' = [a_1', b_1'] \times [a_2', b_2'] \times \ldots \times [a_d', b_d']$ inside $\mathcal{Q}$, where $a_i' = \Delta\lceil a_i/\Delta \rceil$ and $b_i' = \Delta \lfloor b_i/\Delta \rfloor$. If $a_i' \ge b_i'$ for some i, then $\mathcal{Q}'$ is empty. Otherwise, a mode of $\mathcal{Q}'$ is given by $A[rank(\mathcal{Q}', \Delta)]$. Recall that $rank(\mathcal{Q}', \Delta)$ can be computed in constant time when d is a constant. The number of points in the region $\mathcal{Q} \setminus \mathcal{Q}'$ (the region within $\mathcal{Q}$, but outside $\mathcal{Q}'$) is at most $2d\Delta$ (refer to Lemma 1). Then the mode of $\mathcal{Q}$ is either the mode of $\mathcal{Q}'$ or the color of one of the points among the $O(\Delta)$ points in $\mathcal{Q} \setminus \mathcal{Q}'$. Call these $O(\Delta)$ colors the *candidate* colors. Using the range counting structure, for each candidate color c we count the number of points with color c in $\mathcal{Q}$ and report the one with the maximum count. The query time is $O(2d\Delta \cdot t_n) = O(\Delta \cdot t_n)$.

Theorem 2 (Chan et al. [3]) *There exists an $O(s_n + (n/\Delta)^{2d})$-space data structure that supports orthogonal range mode queries on a set of n points in d dimensions in $O(\Delta \cdot t_n)$ time.*

The current best orthogonal range counting data structure requires $s_n = O(n(\log n/\log\log n)^{d-2})$ space and supports queries in $t_n = O((\log n/\log\log n)^{d-1})$ time [7]. The following result can be obtained by choosing Δ such that $s_n = (n/\Delta)^{2d}$. That is $\Delta = n^{(1-\frac{1}{2d})}(\log n/\log\log n)^{(\frac{1}{d}-\frac{1}{2})}$.

Corollary 1 (Chan et al. [3]) *There exists an $O(n(\log n/\log\log n)^{d-2})$-space data structure that supports orthogonal range mode queries on a set of n points in d dimensions in $O(n^{(1-\frac{1}{2d})}(\log n/\log\log n)^{(d+\frac{1}{d}-\frac{3}{2})})$ time.*

4 Improved Data Structure

Again we assume that the input point set $\mathcal{S}$ has been transformed to rank space, and we denote by s_n and t_n the space and query time of an orthogonal range counting data structure on $\mathcal{S}$. The main idea is to maintain the array A in $\Theta((n/\Delta)^{2d})$ bits as opposed to $\Theta((n/\Delta)^{2d})$ words, by using succinct data structures methods. Doing so increases the cost of accessing an entry of A from constant to $O(\Delta \cdot t_n)$ time. The total query cost, however, does not increase.

We now describe how to encode A in less space. We use the following common notation: let $\log^{(h)} n = \log(\log^{(h-1)} n)$ for $h > 1$, let $\log^{(1)} n = \log n$, and let $\log^* n$ be the smallest integer k such that $\log^{(k)} n \le 2$. Let $\Delta_h = \Delta \log^{(h)} n$ (rounded to the next highest power of 2) and let A_h be an array of length $(1+n/\Delta_h)^{2d}$ such that $A_h[i]$ stores the most frequent color in the Δ_h box with $rank(\cdot, \Delta) = i$. Notice that Δ_i is a multiple of Δ_{i+1}, and $\Delta_{\log^* n} = \Theta(\Delta)$.

Lemma 3 *There exists a scheme where A_h can be encoded in $S(h)$ bits and any entry in A_h can be decoded in $T(h)$ time, where*

$$S(h) = \begin{cases} O((n/\Delta_1)^{2d} \log n) & \text{if } h = 1 \\ S(h-1) + O((n/\Delta_h)^{2d} \log^{(h)} n) & \text{if } h > 1, \end{cases}$$

$$T(h) = \begin{cases} O(1) & \text{if } h = 1 \\ T(h-1) + t_n \cdot O(\Delta / \log^{(h)} n) & \text{if } h > 1. \end{cases}$$

Proof. Let A'_h be the desired encoding. The base case can be achieved by storing A_1 explicitly (i.e., $A_1 = A'_1$). For $h > 1$, given an encoding A'_{h-1} we obtain A'_h by storing an additional array B_h of size $(1+\lfloor (n-1)/\Delta \rfloor)^{2d}$ where each entry has size $O(\log^h(n))$ bits. Let R be a Δ_h box and R' be the largest (possibly empty) Δ_{h-1} box within R. We distinguish between two cases:

1. If the mode of R and R' are the same, then we simply store a special symbol \$ in $B_h[rank(R, \Delta_h)]$.

2. Else, there must exists a point p in the region $R \setminus R'$, where $p.c$ is the mode of R. Moreover the distance (say τ) from p to the boundary of R is at most Δ_{h-1}. Then we store $B_h[rank(R, \Delta_h)] = \lceil \tau / \delta_h \rceil$, an approximate value of distance, where $\delta_h = \Delta / \log^{(h)} n$. This approximate distance can be encoded in $O(\log(\Delta_{h-1}/\delta_h)) = O(\log^{(h)} n)$ bits.

Since the space occupied by B_h is $O((n/\Delta_h)^{2d} \log^{(h)} n)$ bits, the equation $S(h) = S(h-1) + O((n/\Delta_h)^{2d} \log^{(h)} n)$ follows.

We now describe how to decode the original value of an entry in A'_h. The array A'_1 is stored explicitly, therefore $T(1) = O(1)$. For $h > 1$, assume that we can decode entries of A'_{h-1} in the desired time. An entry in A'_h corresponding to a Δ_h-box R can be decoded as follows:

1. If $B_h[rank(R, \Delta_h)] = \$$, then the mode of R is same as the mode of R', the largest Δ_{h-1} box within R. The mode of R' is equal to $A_h[rank(R', \Delta_{h-1})]$ so the time for decoding is $T(h) = T(h-1) + O(1)$.

2. Else, $\delta_h \cdot B_h[rank(R, \Delta_h)]$ represents the approximate distance (within an additive error at most $\delta_h = \Delta / \log^{(h)} n$) from a point p from the boundary of R, such that $p.c$ is the mode of R. Since the points are in rank space, the number of points satisfying this approximate distance criteria is at most $2d \cdot \delta_h$ and the color of a point among them is the mode of R. So, the mode of R (i.e., $A_h[rank(R, \Delta_h)]$) can be identified using $O(\delta_h)$ range counting queries. Thus giving the equation: $T(h) = T(h-1) + t_n \cdot O(\Delta / \log^{(h)} n)$.

By combining both cases, the equation $T(h) = T(h-1) + t_n \cdot O(\Delta / \log^{(h)} n)$ follows. $\square$

Note that

$$\begin{aligned} S(\log^* n) &= O\left(\sum_{h=1}^{\log^* n} (n/\Delta_h)^{2d} \log^{(h)} n \right) \\ &= O\left((n/\Delta)^{2d} \sum_{h=1}^{\log^* n} \left(\frac{1}{\log^{(h)} n} \right)^{2d-1} \right) \\ &= O\left((n/\Delta)^{2d} \right), \end{aligned}$$

and

$$\begin{aligned} T(\log^* n) &= t_n \cdot O\left(\sum_{h=1}^{\log^* n} \delta_h \right) \\ &= t_n \cdot O\left(\Delta \sum_{h=1}^{\log^* n} \frac{1}{\log^{(h)} n} \right) \\ &= t_n \cdot O(\Delta). \end{aligned}$$

Therefore, by maintaining an $O((n/\Delta)^{2d})$-bit or $O((n/\Delta)^{2d}/w)$-word data structure structure (along with the range counting structures), we can compute the mode of the largest $\Delta_{\log^* n}$ box $\mathcal{Q}'$ in any query $\mathcal{Q}$ in $t_n \cdot O(\Delta)$ time. Since the number of points in $\mathcal{Q} \setminus \mathcal{Q}'$ is at most $2d \cdot \Delta_{\log^* n} = O(\Delta)$, the mode of $\mathcal{Q}$ can be computed within an additional $O(t_n \cdot \Delta)$ time. We summarize our results in the following theorem.

Theorem 4 *There exists an $O(s_n + (n/\Delta)^{2d}/w)$-space data structure that supports orthogonal range mode queries on a set of n points in d dimensions in $O(\Delta \cdot t_n)$ time.*

We get the following corollary by using the range counting data structure of Jájá et al. [7] with $\Delta = n(w/n)^{(\frac{1}{2d})}(\log n/\log\log n)^{(\frac{1}{d}-\frac{1}{2})}$.

Corollary 2 *There exists an $O(n(\log n/\log\log n)^{d-2})$-space data structure that supports orthogonal range mode queries on a set of n points in d dimensions in $O(n(w/n)^{(\frac{1}{2d})}(\log n/\log\log n)^{(d+\frac{1}{d}-\frac{3}{2})})$ time.*

References

[1] P. Bose, E. Kranakis, P. Morin, and Y. Tang. Approximate range mode and range median queries. In *Proc. STACS*, volume 3404 of *LNCS*, pages 377–388. Springer, 2005.

[2] T. M. Chan, S. Durocher, K. G. Larsen, J. Morrison, and B. T. Wilkinson. Linear-space data structures for range mode query in arrays. In *Proc. STACS*, volume 14 of *Leibniz International Proceedings in Informatics*, pages 291–301, 2012.

[3] T. M. Chan, S. Durocher, K. G. Larsen, J. Morrison, and B. T. Wilkinson. Linear-space data structures for range mode query in arrays. *Theory of Computing Systems*, pages 1–23, 2013.

[4] T. M. Chan, S. Durocher, M. Skala, and B. T. Wilkinson. Linear-space data structures for range minority query in arrays. In *Proc. SWAT*, volume 7357 of *LNCS*, pages 295–306. Springer, 2012.

[5] S. Durocher, R. Shah, M. Skala, and S. V. Thankachan. Linear-space data structures for range frequency queries on arrays and trees. In *Proc. MFCS*, volume 8087 of *LNCS*, pages 325–336. Springer, 2013.

[6] M. Greve, A. G. Jørgensen, K. D. Larsen, and J. Truelsen. Cell probe lower bounds and approximations for range mode. In *Proc. ICALP*, volume 6198 of *LNCS*, pages 605–616. Springer, 2010.

[7] J. JáJá, C. W. Mortensen, and Q. Shi. Space-efficient and fast algorithms for multidimensional dominance reporting and counting. In *Proc. ISAAC*, volume 3341 of *LNCS*, pages 558–568. Springer, 2005.

[8] D. Krizanc, P. Morin, and M. Smid. Range mode and range median queries on lists and trees. *Nordic Journal of Computing*, 12(1):1–17, 2005.

[9] H. Petersen. Improved bounds for range mode and range median queries. In *Proc. SOFSEM*, volume 4910 of *LNCS*, pages 418–423. Springer, 2008.

[10] H. Petersen and S. Grabowski. Range mode and range median queries in constant time and sub-quadratic space. *Information Processing Letters*, 109(4):225–228, 2009.

[11] M. Skala. Array range queries. In *Proc. Space-Efficient Data Structures, Streams, and Algorithms*, volume 8066 of *LNCS*, pages 333–350. Springer, 2013.