The Power of Nonmonotonicity in Geometric Searching*

Bernard Chazelle†

Abstract

We define a close variant of line range searching over the reals and prove that its arithmetic complexity is $\Theta(n \log n)$ if field operations are allowed and $\Theta(n^{3/2})$ if only additions are. This provides the first nontrivial separation between the monotone and nonmonotone complexity of a range searching problem. The result puts into question the widely held belief that range searching for nonisothetic shapes typically requires $\Omega(n^{1+c})$ arithmetic operations, for some constant $c > 0$.

1 Introduction

Conventional wisdom holds that addition is the only arithmetic operation needed for range searching, and that allowing subtraction or multiplication does not help (besides perhaps polylog speedups). Consider line range searching, for example: given n points weighted with real values and n lines in the plane, compute the sum of the weights of the points within each line. This can be done in $\widetilde{O}(n^{4/3})$ time using additions only [12].[1] If other field operations are disallowed (the monotone model), this is essentially optimal, as evidenced by the nearly matching lower bound of $\Omega(n^{4/3})$ established in [5]. Although it is less than clear why subtractions or other field operations should be useful for adding weights, the best lower bound in the nonmonotone model is only $\Omega(n \log n)$ [6]. Our persistent inability to break below $O(n^{4/3})$ for line range searching or, more generally, to achieve $n^{1+o(1)}$ for any typical (nonisothetic) range searching problem [1, 13] has led to the conjecture that it is impossible to do so. Thus, it would appear that the current lower bound technology for the nonmonotone model, which is based on spectral arguments [4, 6, 7] that are inherently unable to produce bounds higher than $\Omega(n \log n)$, is the weak link of the theory.

The results of this paper should lead us to reconsider this belief. We show that allowing subtraction, or more generally multiplication by a constant, adds more power than we might have anticipated. Briefly, we show that if we allow the line to bounce against the walls like a billiard ball, then the problem can be solved in $O(n \log n)$ time in the nonmonotone model (for infinitely many values of n) and $O(n^{3/2})$ time in the monotone case. Both bounds are tight in their respective models. To our knowledge, this is the first example of a nontrivial separation result between the monotone and nonmonotone complexity of a "natural" geometric problem.

---

*This work was supported in part by NSF grant CCR-998817, ARO Grant DAAH04-96-1-0181, and NEC Research Institute.

†Department of Computer Science, Princeton University and NEC Research Institute

[1]The notation $\widetilde{O}(f(n))$ means $= O(f(n) \log^c n)$, for some constant c.

The problem we consider is called *Bouncing-Lines*: Viewing a square grid of n weighted points as a pool table, we define a *query* as the trajectory of a ball shot from any point in an arbitrary direction. The ball bounces off the walls and keeps going forever; the sum of the weights of the points that it encounters along the way constitutes the *answer* to the query. An instance of *Bouncing-Lines* consists of n reals (the weights) and n queries: the output is the set of n answers.

Without the bouncing, the best algorithm known to date runs in time close to $O(n^{4/3})$ time, and this complexity is optimal in the monotone model. What is it about bouncing that makes the complexity drop to $O(n \log n)$? Of course, the complexity of line range searching *without bouncing* might be $O(n \log n)$. If, by chance, we could extend the techniques of this paper to, say, triangle or circular range searching, then it might not be inconceivable that the arithmetic complexity of *every* range searching problem with finite VC dimension is $\widetilde{O}(n)$. This would shatter much of our current understanding in this area. If this is not true, at the very least the results of this paper show that improving the current lower bounds for the nonmonotone model (all of which are based on the spectral lemma or its variant, the trace lemma [4, 6, 7]) is likely to be very challenging.

Our lower bounds also apply to a form of line range searching where lines are considered modulo some integer. While this can be construed as operating on a torus, the problem is not nearly as natural geometrically as *Bouncing-Lines*, which is why we prefer to center our discussion on the latter problem, even though it involves slightly more complicated arguments.

Theorem 1.1 *For an infinite number of values n, the complexity of* Bouncing-Lines *is $\Theta(n \log n)$ time in the nonmonotone model and $\Theta(n^{3/2})$ time in the monotone model.*

Note that, by definition, n is always of the form $(l+1)^2$, where l is the side length of the square grid. The theorem holds for any prime l. The upper bounds are more general, however. The $O(n^{3/2})$ running time is that of the naive algorithm and hence makes no assumption on l. We give a general algorithm for the nonmonotone model which works for any value of l and runs in time $O(d(l)^c\, n \log n)$, where $d(l)$ is the number of divisors of l and c is a positive constant. This implies an upper bound of $O(n \log n)$ for any side length l that is the product of a constant number of primes. It is known [10] that $d(l) < 2^{\log l / \log\log l}$ for all l large enough, and $d(l) < \log l$ for almost all l; in other words, the number of integers $m \le l$ for which $d(m) \ge \log m$ is $o(l)$.

Corollary 1.2 Bouncing-Lines *can be solved in (i) $O(n \log n)$ time for an infinite number of side lengths; (ii) $\widetilde{O}(n)$ for almost all side lengths; and (iii) $n^{1+o(1)}$ for all side lengths.*

Notation: $\mathbf{Z}_n$ denotes the ring of integers modulo n and (m, n) the greatest common divisor of m and n. Euler's totient function, $\varphi(n)$, refers to the number of integers $1 \le k \le n$ such that $(k, n) = 1$. These integers form a multiplicative subgroup of $\mathbf{Z}_n$, which is denoted by $U(\mathbf{Z}_n)$ or simply by U when n is understood. We use the standard notation $d(n)$ and $\omega(n)$ for the number of divisors of n (including 1 and n) and the number of distinct prime factors of n, respectively. (The notation $\omega(n)$ should not be mistaken for its common meaning in computer science.) All number-theoretical facts used in this article can be found in Hardy and Wright's classic text [10]. All logarithms are to the base 2.

2 Bouncing-Line Range Searching

The input consists of the grid of $n = (l+1)^2$ points with integer coordinates in the square $[0,l]^2$. Each grid point p is associated with a real variable $w(p)$, called its *weight.* A query $q = (p, \vec{v})$ is a ray shot from p in direction $\vec{v}$, where p is any point in the square $[0,l]^2$ (not necessarily a grid point). The ray is assumed to bounce against the sides of the square according to the standard reflection law. By convention, if the ray hits a corner, we assume that it bounces back along the same ray but in the opposite direction. The answer to the query q is the sum of the weights of the grid points hit by the (bouncing) ray. Note that a point hit twice or more is counted only once. So, although a query may have infinitely many turns its answer is always finite. *Bouncing-Lines* is the off-line version of this range searching problem: the input consists of n weights and n queries and the output returns the n corresponding sums.

An equivalent formulation is to compute the linear map Aw, where $w \in \mathbf{R}^n$ is the column vector of weights and $A = (A_{ij})$ is the incidence matrix of the underlying set system: $A_{ij} = 1$ if query i passes through point j; and $A_{ij} = 0$ otherwise. To compute the matrix A can be done easily in polynomial time under any standard representation of points and queries, but this is not the point of interest. The focus of our attention is the multiplication of A and w, so we might as well assume that A is already available. For upper bound purposes, we require an algorithm that works on a standard RAM with infinite precision real arithmetic. To prove lower bounds, we limit ourselves to counting the number of arithmetic operations. We view the computation as that of a circuit, where each gate takes two numbers u, v as input and outputs $\alpha u + \beta v$, where α, β are constants associated with the gate (ie, numbers independent of the weight vector w). In the monotone model, the only constants allowed are $\alpha, \beta = 1$. In the nonmonotone version, we relax this assumption and allow α and β to be any complex numbers with bounded moduli.

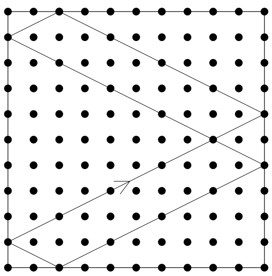

Figure 1: *The bouncing line passes through 19 points.*

3 An Overview of the Algorithm

The basic idea is to define an appropriate duality between points and queries (ie, an incidence-preserving bijection between points and ball trajectories), and endow the points and, by duality, the queries with an abelian group structure. Let p^* denote the dual query of point p. A query q maps back to the point q^*, so that $(p^*)^* = p$. We define a multiplicative abelian group $(G, \otimes)$ over the points, so that given p and p', the point $p \otimes p'$ is another grid point. By isomorphism, we define the dual group $(G^*, \otimes)$ over the queries: the product of two queries $q_1 \otimes q_2$ is defined as $(q_1^* \otimes q_2^*)^*$. Given a point p and a query q, define the characteristic function

$$\chi(p, q) = \begin{cases} 1 & \text{if } p \in q, \\ 0 & \text{else.} \end{cases}$$

The group and the duality must be chosen so as to satisfy the following identity: For any points p, r and query q,

$$\chi(p, q) = \chi(p \otimes r^{-1}, q \otimes r^*). \tag{1}$$

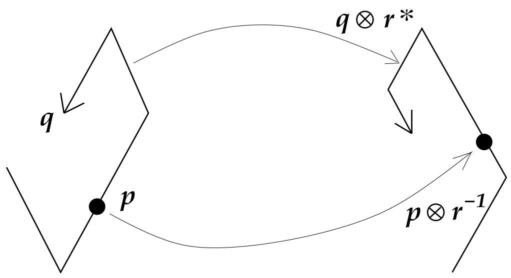

Figure 2: *Group, duality, and incidence.*

If $w(p)$ is the weight associated with point p, then for each query q we must compute

$$W(q) = \sum_p w(p)\chi(p, q) = \sum_p w(p)\chi(\mathbf{1}, q \otimes p^*) = \sum_p \mu(p) g(q^* \otimes p^{-1})$$

where $\mathbf{1}$ denotes the identity element of G, μ is the function $p \mapsto w(p^{-1})$, and g is the function $p \mapsto \chi(\mathbf{1}, p^*)$. It follows that

$$W(q) = (\mu \star g)(q^*),$$

where the convolution $\star$ takes place over the group G. Using the discrete Fourier transform $f \mapsto \widehat{f}$ over G [9], we can write

$$\widehat{W} = \widehat{\mu} \cdot \widehat{g},$$

which means that we can compute $W(q)$ via three Fourier transform computations. (Note the slight abuse of notation in viewing W as a function of q^*.) There exists a Fast Fourier Transform over G^* which allows us to compute the map $f \mapsto \widehat{f}$ in $O(|G^*| \log |G^*|)$ time. We must mention here that the groups in question are not cyclic and that the DFT (not to mention the corresponding FFT) is very different from the sort typically encountered in integer multiplication or in engineering applications; nor is it related to the modular Fourier transform.

Our challenge, therefore, is to define a group G and an appropriate duality which together satisfy (1). Unfortunately, we are not able to do that. The best we can do is partition the set of pairs (*points, queries*) into subsets $(\mathcal{P}_i, \mathcal{Q}_i)$ with respect to which condition (1) holds for some group G_i. Of course, the complexity depends now on the number of such subsets, which naturally we want to keep as small as possible.

4 Preliminaries

To give an algebraic characterization of queries, we consider the group generated by horizontal and vertical reflections that has $[0, l]^2$ as its fundamental domain. This provides a single cover of $\mathbf{R}^2$ that allows us to view any query q as a straightline ray in the plane. Any row of A with a single nonzero entry can be handled separately, so we can assume that every query q passes through at least two grid points. This allows us to express its supporting line ℓ_q as: $v_y(X - p_x) - v_x(Y - p_y) = 0$, where p_x, p_y, v_x, v_y are integers with $(v_x, v_y) = 1$. Any nontrivial element in the group of reflections is of order two, so an equivalent formulation can be given over $\mathbf{Z}_m$, where $m = 2l$: $v_y(X - p_x) - v_x(Y - p_y) = 0 \pmod{m}$. For convenience, we state again the basic relations between grid size, number of points, and modulus:

$$n = (l+1)^2 \qquad \text{and} \qquad m = 2l. \tag{2}$$

Even though v_x and v_y are now reduced modulo m, we can still assume that they are relatively prime. Obviously, their gcd g, if nontrivial, cannot have a common factor with m, and therefore it has an inverse modulo m and so we can use the coefficients v_x/g and v_y/g instead. Note that the queries for which $v_x = 0$ or $v_y = 0$ modulo m can be handled separately in $O(n)$ time, so we restrict ourselves to queries of the following type:

$$\begin{cases} v_y(X - p_x) - v_x(Y - p_y) = 0 \pmod{m} \\ (v_x, v_y) = 1; \quad 0 < v_x, v_y < m\,. \end{cases} \tag{3}$$

We verify that any equation of type (3) corresponds to the trajectory of a bouncing ball. To see why, because $(v_x, v_y) = 1$, $iv_x - jv_y = 1$ for some $i, j \in \mathbf{Z}$, and so any (X, Y) that satisfies (3), ie, $v_y(X - p_x) - v_x(Y - p_y) = km$, for some k, maps injectively to the point $(X + jkm, Y + ikm)$, which is in the same residue class and lies on the real line $v_y(X - p_x) - v_x(Y - p_y) = 0$. Note that this might not be true if v_x and v_y have a common factor: for example, the point $(2, 0)$ on the line $2X + 2Y = 0 \pmod 4$ is not in the residue class of any point on the corresponding real line.

Note that there are enough distinct lines to make the problem interesting. For example, set $v_x = 1$ and $p_x = 0$, and observe that (3) becomes $Y = v_y X + p_y \pmod{m}$, which gives us $\Omega(n)$ distinct lines.

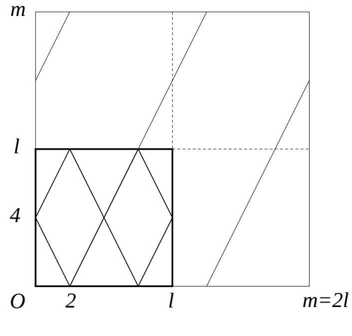

Figure 3: *A bouncing line and its line* $2X - Y = 4 \pmod{16}$.

Consider a line of type (3) and rewrite it as $v_y X + v_x Y = v_0 \pmod m$, with $(v_x, v_y) = 1$ and $0 < v_x, v_y < m$ (note that we changed v_x into $-v_x$ for notational convenience). An even better characterization of the line can be obtained as follows: Let g be the largest divisor of m that is relatively prime to v_x, and let $f = m/g$. Note that f may not be equal to (v_x, m). From the definition of g and the fact that $(v_x, v_y) = 1$, it follows that $(v_x, g) = (v_y, f) = 1$. Since $(f, g) = 1$, we can use the Chinese remainder theorem (CRT) to derive an equivalent formulation of the line:

$$\begin{cases} X + f_x Y = f_0 \pmod f \\ g_y X + Y = g_0 \pmod g . \end{cases} \tag{4}$$

If $f = 1$ (resp. $f = m$) then, of course, the first (resp. second) equation is trivial. Note that (X, Y) takes on f values modulo f and g values modulo g. By the CRT, it follows that any line has exactly $fg = m$ points.

Lemma 4.1 *Any line of type* (3) *contains exactly* m *points.*

One final word about the modeling of the problem in terms of modular arithmetic. Figure 1 indicates that the ball passes through 19 points, but $l = 10$ and so we should expect 20 of them. Where is the missing point? The problem is that $(8, 5)$ is a multiple point. By convention, the weight should be counted only once but if we solve the problem modulo m it will be counted twice. We repair the damage by identifying all the multiplicities (ie, pairs of point-line incidences for (3) whose images within $[0, l]^2$ coincide), and subtracting each weight accordingly.

Lemma 4.2 *The number of multiplicities is* $O(2^{\omega(m)} m^2 d(m))$.

Proof: See appendix. □

To find the multiple points is something we need to do when setting up the matrix A. It requires standard linear algebra. Again, note that this does not concern the circuit complexity of the problem itself, so we may skip the details. What we have shown is that, with the help of $O(2^{\omega(m)} m^2 d(m))$ subtractions performed in a postprocessing step to correct multiple weight counts, *Bouncing-Lines* can be reduced to the following problem: Given a column vector $w \in \mathbf{R}^{m^2}$, compute Aw, where A is the m^2-by-m^2 incidence matrix of a set system defined by points and lines of type (3), where $A_{ij} = 1$ if line i contains point j and 0 otherwise.

5 A Fast Algorithm

To define a group structure on points and queries, we single out the points with appropriate number-theoretic properties. We say that a point (x, y) is *polar* if both x and y belong to the multiplicative group U of units of $\mathbf{Z}_m$. A line is said to be polar if it can be expressed by an equation of the form $aX + bY = 2^j \pmod m$, where $a, b \in U$ and $0 \le j \le e_1$, with 2^{e_1} being the largest power of two dividing m. When j is specified, we say that the line is j-polar: j is called the *polarization type.* Whereas the notion of polarity is essential to the identification of group structures among points and lines, the polarization type is introduced purely for technical reasons.

Let $\mathcal{P}$ be the set of grid points $[0, m-1]^2 \cap \mathbf{Z}^2$ and let $\mathcal{L}$ be the set of all distinct lines with coefficients modulo m; write $\Sigma = \mathcal{P} \times \mathcal{L}$. We say that a subset of Σ is *polarized* if it is a Cartesian product that can be affinely brought into polar form, ie,

(i) the subset is of the form $P \times L$, where $P \subseteq \mathcal{P}$ and $L \subseteq \mathcal{L}$; and

(ii) there exist a nonsingular affine transformation f and an integer j such that $f(p)$ is polar for each $p \in P$ and $f(\ell)$ is j-polar for each $\ell \in L$.

Lemma 5.1 *Let 2^{e_1} be the largest power of two dividing m. The set Σ can be partitioned into at most $(e_1+1)c^{\omega(m)}$ polarized sets, for some constant $c > 0$.*

Proof: For the time being, assume that m is a prime power q^e, where $q > 2$. In the expression of the line $v_y X + v_x Y = v_0 \pmod{m}$ provided by (4), it is immediate that either f or g is equal to m. Without loss of generality, assume that $f = m$. The equation of the line becomes

$$X + f_x Y = f_0 \pmod{m}. \tag{5}$$

Consider the affine transformation $f_{s,t,u}(p)$, where $p = (x, y) \in \mathbf{Z}_m^2$ and

$$f_{s,t,u}(p) = \begin{cases} x' = y + u \pmod{m} \\ y' = sy + x + t \pmod{m}. \end{cases} \tag{6}$$

The transformation is nonsingular and so the line can be expressed equivalently as

$$f_{s,t,u}(\ell):\ (f_x - s)X' + Y' = f_0 + (f_x - s)u + t.$$

For any point/line pair (p, ℓ), where $p = (x, y)$ lies on a line ℓ of type (5), there exist some $s, u \in \{0, 1\}$ and $t \in \{0, 1, 2\}$ such that both $f_{s,t,u}(p)$ and $f_{s,t,u}(\ell)$ are polar. To see why, consider the following assignments of s, t, u: If $y = 0 \pmod{q}$, then $u = 1$, else $u = 0$. If $f_x = 0 \pmod{q}$, then $s = 1$, else $s = 0$. For any fixed x, y, f_0, f_x, s, u, each one of the equations (in t) $sy + x + t = 0 \pmod{q}$ and $f_0 + (f_x - s)u + t = 0 \pmod{q}$ has exactly one solution. Because $q > 2$, this implies that neither equation is satisfied for at least one value of $t \in \{0, 1, 2\}$. The point $f_{s,t,u}(p)$ is polar and the line $f_{s,t,u}(\ell)$ has for equation, $aX' + Y' = b \pmod{m}$, where $a, b \in U$. Multiplying it by the inverse of b modulo m shows that the line is 0-polar. This proves our claim that the 12 transformations considered are sufficient to bring every pair (p, ℓ) into polar form. Obviously, the subset $\Sigma_{s,t,u}$ of Σ that is brought into polar form by a given $f_{s,t,u}$ is a rectangle, ie, a set of the form $P \times L \subseteq \Sigma$. This shows that Σ can be expressed as a (nondisjoint) union of at most 12 polarized sets $\Sigma_{s,t,u}$.

We turn the cover of Σ into a partition in the obvious way: we mark every point of $\mathcal{P}$ (resp. line of $\mathcal{L}$) with a 12-bit vector indicating which of the 12 values of (s, t, u) make the point (resp. line) polar (resp. 0-polar). This partitions $\mathcal{P}$ and $\mathcal{L}$ into subsets: for every pair of them, $P \subseteq \mathcal{P}$, $L \subseteq \mathcal{L}$, whose bit vectors have a common intersection, we form the product $P \times L$. These products are obviously disjoint; by the previous argument, they in fact partition all of Σ. We can repeat the same argument for the case $g = m$. To summarize, we have shown that if m is a prime power q^e for $q > 2$, then Σ can be partitioned into no more than $c \leq 2(2^{12})^2$ polarized sets $P_i \times L_i$, each one associated with a nonsingular affine transformation f_i. (No effort has been made to optimize the constant c.)

The case $q = 2$, ie, $m = 2^e$, is the reason why we added a 2^j term to the definition of a polar line. The previous argument breaks down, as no single value of t can always ensure that both $sy + x + t$ and $f_0 + (f_x - s)u + t$ are odd. However, we can always enforce, say, the first condition. The difference in outcome is that the partition of Σ enjoys slightly weaker properties. A line $\ell \in L_i$ is such that $f_i(\ell)$ has an equation of the form $aX' + Y' = b \pmod{m}$, where a is odd but $b < m$ is arbitrary. If b has an odd factor, we can always multiply the equation by its inverse to put it in polar form. Note that the case $b = 0$ corresponds to a polarization type equal to e.

We now consider the general case where m is an arbitrary positive integer. Let $\prod_{i=1}^{k} p_i^{e_i}$ be the prime factor decomposition of m. Note that m is always even, so $p_1 = 2$. The ring homomorphism of the CRT allows us to classify point-line incidences one modulus at a time. Points and lines are incident if and only if they are so modulo each $p_i^{e_i}$. Regarding polarity, the same of is true of points: $p = (x, y)$ is polar if and only if the point $(x \bmod p_i^{e_i},\ y \bmod p_i^{e_i})$ is polar for each i. The case of lines requires a brief discussion. A line $aX + bY = d \pmod{m}$ is polar if and only if it is polar modulo each $p_i^{e_i}$.

One direction is easy: take a polar line $aX + bY = 2^j \pmod{m}$, with $a, b \in U$ and $0 \leq j \leq e_1$. It is obviously j-polar modulo $p_1^{e_1}$. For any odd p_i, observe that 2^j is relatively prime to $p_i^{e_i}$, and therefore has an inverse modulo $p_i^{e_i}$. Thus the line's equation can be rewritten as $a'X + b'Y = 1 \pmod{p_i^{e_i}}$, with a', b' relatively prime to $p_i^{e_i}$, showing that it is 0-polar.

Conversely, suppose that the line $aX + bY = d \pmod{m}$ is j-polar modulo $p_1^{e_1}$ and 0-polar modulo $p_i^{e_i}$, for any $i > 1$. No p_i can divide a, and therefore $a \in U$; same with b. No odd p_i divides d; therefore $d = 2^j + \alpha 2^{e_1} = 2^j \beta$, where $\beta = \alpha 2^{e_1 - j} + 1$. If $j < e_1$, then β is odd. This implies that $\beta \in U$; therefore, after dividing by β, the line can be expressed as $a'X + b'Y = 2^j \pmod{m}$, with $a', b' \in U$, thus showing that it is j-polar. If $j = e_1$, then $d = 2^{e_1} 2^{\delta} \gamma$, for $\delta \geq 0$ and odd γ relatively prime to $m/p_1^{e_1}$. Let $h < m$ satify

$$\begin{cases} h = 1 \pmod{p_1^{e_1}} \\ h 2^{\delta} \gamma = 1 \pmod{m/p_1^{e_1}}. \end{cases}$$

By the CRT, h exists and is unique. Obviously $h \in U$; so multiplying by h gives the equivalent line equation, $a'X + b'Y = 2^{e_1} h 2^{\delta} \gamma \pmod{m}$, with $a', b' \in U$, ie,

$$a'X + b'Y = 2^{e_1}(1 + jm/p_1^{e_1}) = 2^{e_1} \pmod{m},$$

which again proves that the line is j-polar. For illustration, here is an example with $m = 20$: The line $X + Y = 4 \pmod{4}$ is 2-polar modulo 4, while $X + Y = 1 \pmod{5}$ is 0-polar modulo 5. This gives us the line $X + Y = 16 \pmod{20}$, which after multiplication by 9 becomes $9X + 9Y = 4 \pmod{20}$, which indeed proves that the line is 2-polar modulo m.

Thus, we can generalize the previous result by simply taking the product of the partitions defined over the moduli. Taken modulo $p_i^{e_i}$, Σ is partitioned into no more than c polarized sets $P_l^i \times L_l^i$ (polarization defined modulo $p_i^{e_i}$), each one associated with a nonsingular affine transformation f_l^i. By the CRT, this induces a partition of Σ modulo m into at most c^k polarized sets of the form $\prod_{i=1}^{k} P_{l_i}^i \times L_{l_i}^i$. By this notation we mean to include any pair (p, ℓ), where p and ℓ belong respectively to $P_{l_i}^i$ and $L_{l_i}^i$ modulo $p_i^{e_i}$. For each subset $\prod_{i=1}^{k} P_{l_i}^i \times L_{l_i}^i$, polarization is defined modulo m. Its associated transformation $f_{l_1,\dots,l_k} = (f^1_{l_1}, \dots, f^k_{l_k})$ is defined by specifying its reduction $f^i_{l_i}$ modulo $p_i^{e_i}$, ie, $f_{l_1,\dots,l_k}(p) = (x, y)$ where $f^i_{l_i}(p) = (x, y) \pmod{p_i^{e_i}}$ for each $1 \le i \le k$. Similarly, $f_{l_1,\dots,l_k}(\ell)$ is the unique line modulo m whose coefficients modulo $p_i^{e_i}$ specify the line $f^i_{l_i}(\ell)$.

The partition of Σ satisfies the lemma, except for the condition that the polarization of the lines within each subset be defined with respect to a unique type j. Each $P^1_l \times L^1_l$ can be partitioned into $P^1_l \times L^1_{l,j}$, for $j = 0, 1, \dots, e_1$, to distinguish among all possible polarization types. This adds a factor of $e_1 + 1$ to the size of the partition. Noting that $k = \omega(m)$, the proof is complete. □

Given an input set of m^2 points and m^2 lines, let $\{P_i \times L_i\}$ be the partition of the set of line/point pairs (ℓ, p) induced by Lemma 5.1. Recall that the problem is equivalent to computing Aw, where w is the column vector of weights and A is the m^2-by-m^2 incidence matrix: $A_{ij} = 1$ if line i contains point j and 0 otherwise. Let B_i be the m^2-by-m^2 incidence matrix corresponding to $P_i \times L_i$. Obviously, $A = \sum_i B_i$, so it suffices to explain how to compute $B_i x$. Let f_i be the nonsingular affine transformation associated with $P_i \times L_i$ and let j_i be its polarization type. By definition, for any $p \in P_i$ and $\ell \in L_i$, both coordinates of $f_i(p)$ lie in U and the line $f_i(\ell)$ is expressed by the equation

$$aX + bY = 2^{j_i} \pmod{m}, \tag{7}$$

where $a, b \in U$ and $0 \le j_i \le e_1$, with 2^{e_1} being the largest power of two dividing m.

We endow U^2 with an abelian multiplicative group structure by looking at it as the direct product $U \otimes U$. In other words, if $p = (x, y)$ and $p' = (x', y')$ belong to $U \otimes U$, then $p \otimes p' = (xx', yy') \in U \otimes U$. For a fixed polarization type, here j_i, we define the duality

$$\left\{ \begin{array}{lll} p = (x, y) & \mapsto & p^* : xX + yY = 2^{j_i} \pmod{m} \\ \ell : aX + bY = 2^{j_i} \pmod{m} & \mapsto & \ell^* = (a, b)\,. \end{array} \right. \tag{8}$$

Note that lines are treated here as formal expressions, not as subsets of $\mathbf{Z}_m$. For example, the two lines $X + Y = 2 \pmod{12}$ and $7X + 7Y = 2 \pmod{12}$ are considered distinct, even though they contain the same set of points. Next, we express the map B_i as a convolution. The sum of weights along line ℓ is given by

$$W_i(\ell) = \sum_{p \in U \otimes U} w_i(p) \chi_\ell(p),$$

where $w_i(p)$ is the weight of point p if it belongs to P_i and 0 otherwise, and $\chi_\ell(p)$ is 1 if $p \in \ell$ and 0 otherwise. If we define $g(p)$ to be 1 if $x + y = 2^{j_i} \pmod{m}$ and 0 otherwise, for $p = (x, y)$, then $\chi_\ell(p) = g(\ell^* \otimes p)$. Writing $\mu_i(p) = w_i(p^{-1})$ for any $p \in U \otimes U$, we find that

$$W_i(\ell) = \sum_{p \in U \otimes U} \mu_i(p) g(\ell^* \otimes p^{-1}).$$

This shows that W_i is the convolution $\mu_i \star g$. Harmonic analysis over finite groups gives us the tools to diagonalize this convolution. By the structure theorem for finite abelian groups [2], $U \otimes U$ is isomorphic to a direct product of cyclic groups $U_1, \dots, U_\tau$ of order $d_1 \le \dots \le d_\tau$, with $d_1 | d_2 | \cdots | d_\tau$. To factor $U \otimes U$ in this way is easily done in polynomial time. (Again, this takes place in preprocessing, so this overhead is not added to the cost of the range searching problem.) Obviously, it suffices to factor U.

To do that, we identify an element of highest order, factor out the corresponding cyclic group and then repeat this process with respect to the quotient group. This factors U into cyclic groups. It is easy to see that picking only elements of highest order ensures that the orders of the factors form a descending chain of divisors.

The Fourier transform of a function $f : x \in U \otimes U \mapsto \mathbf{R}$ is defined by specifying the group of characters for $U \otimes U$. The factorization of $U \otimes U$ allows us to identify $x \in U \otimes U$ with a vector $(x_1, \ldots, x_\tau)$, where $x_i \in \mathbf{Z}_{d_i}$, with the group operation achieved by coordinate-wise addition. The $|U|^2$ characters are:

$$t = (t_1, \ldots, t_\tau) \in U \otimes U \quad \mapsto \quad e^{2\pi i (t_1 x_1/d_1 + \cdots + t_\tau x_\tau / d_\tau)}.$$

We define the Fourier transform of f and its inverse:

$$\widehat{f}(t) = \sum_{x \in U \otimes U} f(x) e^{-2\pi i \sum_{j=1}^{\tau} t_j x_j / d_j} \qquad \text{and} \qquad f(x) = |U|^{-2} \sum_{t \in U \otimes U} \widehat{f}(t) e^{2\pi i \sum_{j=1}^{\tau} t_j x_j / d_j}.$$

The convolution formula gives

$$\widehat{W_i}(t) = \widehat{\mu_i}(t)\, \widehat{g}(t).$$

So, we can derive W_i by computing two Fourier transforms and one inverse. The classical FFT algorithm has been generalized to arbitrary abelian groups [3, 8, 11], and it is possible to compute it in $O(N \log N)$ time, where N is the order of the group. This shows that $B_i x$ can be computed in $O(m^2 \log m)$ time; therefore, by Lemma 5.1, Ax can be derived in time $O(e_1 c^{\omega(m)} m^2 \log m)$. Counting the preprocessing of Lemma 4.2, the time for solving an instance of *Bouncing-Lines* is $O(e_1 c^{\omega(m)} m^2 \log m + 2^{\omega(m)} m^2 d(m))$. Obviously, $e_1 c^{\omega(m)} \leq d(m)^{1+\log c}$; therefore, by (2),

Theorem 5.2 Bouncing-Lines *can be solved in* $O(d(l)^b\, n \log n)$ *time, for some constant* $b > 0$, *where* $n = (l+1)^2$.

If only additions are allowed, then the naive algorithm, which involves adding the weights one at a time, has a complexity of $O(n^{3/2})$. This establishes the upper bounds of Theorem 1.1 and Corollary 1.2.

6 Lower Bounds

It remains for us to prove the two lower bounds of Theorems 1.1. We begin with the $\Omega(n^{3/2})$ bound for the monotone case. The idea is to build a problem instance whose incidence matrix does not have large rectangles of ones. The lower bound follows from a result of [5].

Assume that l is prime. We take as input all the lines of the form $Y = aX + b$, for all $0 < a, b < l/2$, and all points of $[0, l]^2 \cap \mathbf{Z}^2$. By (2) this produces a set system with $\Omega(n)$ queries and $\Omega(n)$ points. Given two distinct queries $Y = aX + b \pmod m$ and $Y = a'X + b' \pmod m$, how many points can they share? Accounting for reflections, any intersection point (x, y) satisfies:

$$\begin{cases} Y = aX + b \pmod m \\ sY = ta'X + b' \pmod m, \end{cases}$$

for some $s, t \in \{-1, 1\}$. This implies that $(a - sta')X = sb' - b \pmod m$. Looking at that equation modulo l, we distinguish between two cases:

- $a \neq sta'$ (mod l): then X is uniquely specified modulo l, and by the CRT has no more than two solutions modulo m. This implies at most two intersection points.
- $a = sta'$ (mod l): because $0 < a, a' < l/2$, it must be the case that $st = 1$ and $a = a'$. Note that the case $s = t = -1$ is impossible because it would imply that $b + b' = 0$ (mod l), with $0 < b, b' < l/2$. Therefore, $s = t = 1$ and hence $b = b'$. This contradicts the assumption that the queries are distinct.

If the two queries are distinct, then they share at most $2|\{ (s,t) \}| = 8$ points. Now, switching over to the set system formed by the bouncing lines, the incidence matrix A has $\Omega(m^2)$ rows and columns, and its number of ones is $\Omega(m^3)$. Furthermore, it contains no 2-by-9 rectangles of ones. By [5, 7], it follows that the monotone complexity of computing Ax is $\Omega(m^3) = \Omega(n^{3/2})$.

To deal with the nonmonotone case where all field operations are allowed, we use the *trace lemma* [6, 7]. For completeness, we state it below:

Lemma 6.1 [6] *The circuit complexity of $x \mapsto Ax$, where A is an N-by-N incidence matrix is*

$$\Omega_\varepsilon\Big(N \log\Big(\operatorname{tr} M/N - \varepsilon\sqrt{\operatorname{tr} M^2/N}\,\Big)\Big),$$

where $M = A^T A$ and $\varepsilon > 0$ is an arbitrarily small constant.

The proof of the trace lemma was proven in a circuit model allowing only additions and subtractions. The same proof, however, can easily accommodate gates of the form $(u, v) \in \mathbf{C} \mapsto \alpha u + \beta v$, where α and β are complex numbers with bounded moduli. In this way, the model allows the use of Fourier transforms. In the problem at hand, we have $N = \Theta(m^2)$ and $\operatorname{tr} M = \Theta(m^3)$. Every pair of distinct rows in A gives rise to at most 8^2 two-by-two submatrices of ones; on the other hand, each row and each column has $O(l)$ ones. The trace of M^2 is the number of rectangles of ones (ie, j-by-k submatrices of ones, with $1 \leq j, k \leq 2$); it follows that $\operatorname{tr} M^2 = O(m^4)$. By Lemma 6.1, the complexity of computing Ax is at least proportional to $m^2 \log((c - \varepsilon c')m)$, for some constants $c, c' > 0$. Setting ε small enough completes the proof of the lower bounds of Theorem 1.1.

7 Conclusion

A similar separation result can be obtained for line range searching on a torus and higher-dimensional variants of *Bouncing-Lines*. To determine whether such a separation between monotone and nonmonotone complexity can be found for more standard range searching, eg, orthogonal ($n \log\log n$ vs. $n \log n$) or circular & nonisothetic ($n \log n$ vs. n^{1+c}), is an outstanding open problem. Whether this is the case or not, this work makes it likely that improving the lower bounds derived from the spectral and trace lemmas will prove either impossible or at least extremely challenging.

8 Acknowledgments

I wish to thank the anonymous referees for their helpful comments and suggestions regarding the presentation of the results.

Appendix

Proof of Lemma 4.2 A multiple point (x, y) of line (3) is such that, modulo m, $(x, -y)$, $(-x, y)$, or $(-x, -y)$ belongs to the line. Note that the multiplicity can be as high as 4: for example, take the point $(3, 2)$ on the line $2X + 3Y = 0 \pmod{12}$. We count the number of pairs (p, ℓ) for all points and lines, where $p = (x, y)$ is a multiple point of line ℓ. By symmetry it suffices to consider the two cases: (i) $(x, y), (-x, y) \in \ell$ and (ii) $(x, y), (-x, -y) \in \ell$.

(i) Using the characterization (4), it follows that $2f_x y = 2f_0 \pmod{f}$, $2x = 0 \pmod{f}$, $2y = 2g_0 \pmod{g}$, and $2g_y x = 0 \pmod{g}$. The number of pairs f, g is at most $2^{\omega(m)}$, while the number of pairs f_0, g_0 is equal to fg. For fixed f_0, g_0, there are at most $4fd(f)$ pairs of the form f_x, y and, similarly, $4gd(g)$ pairs of the form g_y, x. We briefly explain why in the case of f_x, y, the other case being similar. We shall use the fact that the equation $2y = 2g_0 \pmod{g}$ has at most two solutions; therefore, by the CRT, fixing y modulo f leaves at most two possibilities for y modulo m.

- If f is odd, then $f_x y = f_0 \pmod{f}$. Fix a divisor f' of f and count the number of pairs f_x, y such that $(f_x, f) = f'$. Since f' must divide f_0, we have $(f_x/f')y = f_0/f' \pmod{f/f'}$. This means that y is unique modulo f/f', and hence one of f' values modulo f and one of at most $2f'$ values modulo m (the factor 2 comes from the fact that y has at most two solutions modulo g, for fixed g_0). There are at most f/f' choices for f_x, which gives a total count of $2f'(f/f')$ possibilities for the pair f_x, y. Summing over all f', this puts an upper bound of $2fd(f)$ on the number of pairs f_x, y.
- If f is even, then $f_x y = f_0 \pmod{f/2}$. Now, we fix a divisor f' of $f/2$ and count the number of pairs f_x, y such that $(f_x, f/2) = f'$. It follows that $(f_x/f')y = f_0/f' \pmod{f/2f'}$, and so y is unique modulo $f/2f'$, and hence one of $2f'$ values modulo f and $4f'$ values modulo m. This yields at most f/f' choices for f_x, which produces a count of $4f'(f/f')$ possibilities for the pair f_x, y. This time, we have an upper bound of $4fd(f)$ on the number of pairs f_x, y.

With our claim now proven, we see that the number of 8-tuples $(f, g, f_0, g_0, f_x, g_y, x, y)$ is bounded by $O(2^{\omega(m)} f^2 g^2 d(f) d(g))$. The function d is multiplicative (ie, $d(x)d(y) = d(xy)$ for relatively prime x, y), and so the number of multiplicities of type (i) is $O(2^{\omega(m)} m^2 d(m))$.

(ii) We now have $2f_0 = 0 \pmod{f}$ and $2g_0 = 0 \pmod{g}$. The number of pairs (f_0, g_0) is at most 4. For fixed f, g, f_0, g_0, and fixed $x \pmod{f}$ and $y \pmod{g}$, the previous argument shows that the numbers of pairs (f_x, y) and (g_y, x) are bounded by $2fd(f)$ and $2gd(g)$, respectively. This gives a total of at most $O(2^{\omega(m)} m^2 d(m))$ multiplicities of type (ii).

□

References

[1] Agarwal, P.K., Erickson, J. *Geometric range searching and its relatives*, in "Advances in Discrete and Computational Geometry," eds. Chazelle, B., Goodman, J.E., Pollack, R., Contemporary Mathematics 223, Amer. Math. Soc., 1999, pp. 1–56.

[2] Artin, M. *Algebra*, Prentice Hall, 1991.

[3] Baum, U., Clausen, M., Tietz, B. *Improved upper complexity bounds for the discrete Fourier transform*, Applicable Algebra in Engineering, Communication and Computing 2 (1991), 35–43.

[4] Chazelle, B. *A spectral approach to lower bounds with applications to geometric searching*, SIAM J. Comput. 27 (1998), 545–556.

[5] Chazelle, B. *Lower bounds for off-line range searching*, Disc. Comput. Geom. 17 (1997), 53–65.

[6] Chazelle, B., Lvov, A. *A trace bound for the hereditary discrepancy*, Disc. Comput. Geom. 26 (2001), 221–231.

[7] Chazelle, B. *The Discrepancy Method: Randomness and Complexity*, Cambridge University Press (2000); paperback edition, 2001.

[8] Clausen, M. *Fast Fourier transforms for metabelian groups*, SIAM J. Comput. 18 (1989), 584–593.

[9] Dym, H., McKean, H.P. *Fourier Series and Integrals*, Probability and Mathematical Statistics, Vol. 14, Academic Press, 1972.

[10] Hardy, G.H., Wright, E.M. *An Introduction to the Theory of Numbers*, 5th ed., Oxford Science Publications, 1979.

[11] Maslen, D.K., Rockmore, D.N. *Generalized FFTS - A survey of some recent results*, in "Groups and Computation II", eds. L. Finkelstein and W.M. Kantor, DIMACS Series in Disc. Math. and Theoret. Comput. Sci. 28 (1997), 183–237.

[12] Matoušek, J. *Range searching with efficient hierarchical cuttings*, Disc. Comput. Geom. 10 (1993), 157–182.

[13] Matoušek, J. *Geometric range searching*, ACM Comput. Surv. 26 (1994), 421–461.