Pattern Minefield Detection from Inexact Data*

Gabriel Robins and Brian L. Robinson†

Department of Computer Science, University of Virginia, Charlottesville, VA 22903
† American Management Systems, Inc., Arlington, VA 22209

Extended Abstract

Detecting spatial regularity in images arises in military applications, computer vision, scene analysis, and other areas. In this paper we give a $O(n^{5/2})$-time algorithm that for a given pointset finds all maximal approximately equally-spaced collinear subsets. Our algorithm is robust in that it can tolerate noise or imprecision that may be inherent in the measuring process. Thus, our algorithm is applicable in real-world arenas, such as landmine detection from infrared images.

Keywords: Landmine detection, spatial regularity, computational geometry, algorithms.

1 Introduction

Spatial regularity detection is an important problem in a number of domains such as computer vision, scene analysis, and landmine detection [3]. This paper focuses on the spatial regularity problem in the context of landmine detection from infrared terrain images, where for a given set of points, the goal is to recognize equally-spaced collinear subsets. The detection of landmines from infrared images is made possible by two facts [5]: (1) landmines are usually deployed equally-spaced along straight rows, in order to enable their future recovery; and (2) landmines have a distinct signature in infrared images obtained from airborne or orbiting platforms.

Kahng and Robins [3] gave an optimal $O(n^2)$-time algorithm for the *exact* version of the landmine detection problem, and their algorithm was later parallelized by Boxer and Miller [1]. However, we must often work with *inexact* data due to inaccuracies in mine placement, drift due to soil shifting, and imprecision inherent in the imaging technology. Since these factors can corrupt the collinearity and equal-spacedness properties that we seek, exact methods which do not compensate for these phenomena may be inadequate in practice.

We address this issue by developing a robust algorithm for mine detection that can tolerate inaccuracies in the data relative to a user-specified error threshhold. In particular, for an input pointset our algorithm will report all equally-spaced collinear subsets, where points are allowed to drift from their ideal locations by up to a given amount. Our algorithm relies on computational geometric and combinatorial techniques, and runs within $O(n^{5/2})$ time.

*Professor Gabriel Robins is supported by National Science Foundation Young Investigator Award MIP–9457412. Correspondence should be directed to Professor Gabriel Robins, Department of Computer Science, Thornton Hall, University of Virginia, Charlottesville, VA 22903, USA, Email: robins@cs.virginia.edu, phone: (804) 982-2207, FAX: (804) 982-2214, WWW URL http://www.cs.virginia.edu/~robins/

2 Preliminaries

Given a set of distinct points $P \subset E^2$, a subset $P' \subseteq P$ is *collinear* if all points of P' lie on the same line. We say that a collinear subset $P' \subseteq P$ is *equally-spaced* if all points of P' are equally-spaced along their containing line. A sequence of points that is both collinear and equally-spaced is said to be *regular*.

Some techniques for addressing imprecision in geometric problems were introduced by Guibas et al. in [2]. While these methods are not readily applicable to our problem, we adopt some of their notation. In particular, a pointset is *ε-regular* if each point may be displaced by a distance of at most ε to yield a regular pointset; i.e. given a fixed $\varepsilon \geq 0$, a pointset $P = \{p_1, p_2, \ldots, p_n\} \subset E^2$ is ε-regular if there exists a regular pointset $R = \{r_1, r_2, \ldots, r_n\}$ such that $dist(p_i, r_i) \leq \varepsilon$, for all $1 \leq i \leq n$ (see Figure 1), where $dist(a, b)$ denotes the distance between points a and b.

A *maximal* set with respect to a particular property (e.g., collinearity, regularity, near-regularity, etc.) is one that is not properly contained in any other set having the same property. A *sequence* denotes an ordering of the points of an ε-regular subset based on the ordering of their corresponding counterparts in the regular sequence. We may now formulate our problem as follows:

Mine Detection Problem (MDP): Given $\varepsilon \geq 0$ and a pointset $P \subset E^2$, find all maximal ε-regular sequences contained in P.

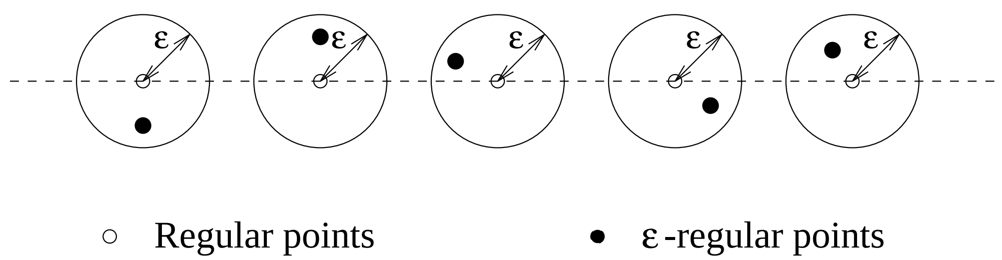

Figure 1: An example of an ε-regular pointset (solid dots), whose points are within ε of the corresponding points of a regular pointset (hollow dots).

We can show the following lower bound on the time complexity of the Mine Detection Problem:

Theorem 2.1 *The time complexity of the mine detection problem has a lower-bound of $\Omega(n^2)$ for all $\varepsilon \geq 0$.*

Unfortunately the general MDP problem formulation above is intractable since the output can have exponential size, as in the example of Figure 2: for each of the points in the regular set (hollow dots) there are two observed points (solid dots) within a distance of ε. Thus, a maximal ε-regular subset may be constructed by choosing either of the two solid dots in each circle, which induces an exponential number of distinct maximal ε-regular subsets.

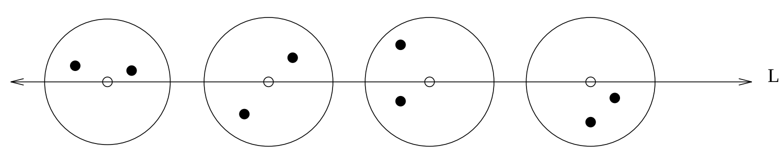

Figure 2: An example of an instance with exponential output.

Another obstacle is the non-linearity of the Euclidean norm formula, which gives rise to second-degree optimization constraints. In light of these considerations, we propose the following modifications to the MDP formulation above in order to enable a polynomial-time algorithm:

- First, to keep the equally-spaced constraint equations linear (and thus easier to optimize), we employ the Chebyshev metric[1] for measuring distances between points in the ε-regular sequence and the corresponding points in the associated regular sequence; and
- Second, to prevent an exponential-sized output, we postulate that each interpoint distance in the input data is greater than 8ε.

The first modification implies that the circles in Figure 1 become squares. Because a square of side 2ε can cover a circle of diameter 2ε, no potential mine locations will be overlooked given this modification. On the other hand, a few "false positives" may be detected, but this is in practice preferable to landmines remaining undiscovered. In the rest of the discussion we assume that all distances are Chebyshev unless otherwise noted.

The second modification relating the magnitudes of mine seperation and potential drift is not unrealistic, since the sizes of mines and their potential drift tend to be considerably smaller than typical inter-mine separation. As we shall see below, these two modifications enable a practical, efficient solution. Moreover, even if the input does not satisfy the second condition, our proposed algorithm will still produce a correct solution, although a polynomial time complexity is no longer guaranteed in such cases.

3 A Robust Algorithm for Mine Detection

The high-level description of the algorithm is as follows. Given a pointset, our basic approach is to extend each pair of points into a maximal ε-regular sequence. At each step of the algorithm we seek a single additional point which may be included into the current sequence while preserving the sequence's ε-regular property. When we find such a point we *extend* the sequence by including the point into the sequence. When a maximal ε-regular sequence has been found through repeated extensions, we iterate this process with all other pairs of points.

Since any contiguous subsequence of an ε-regular sequence is itself ε-regular, we do not need to reconstruct any subsequence which is a contigous portion of a previously found maximal ε-regular sequence. Specifically, when starting with a pair, we repeatedly extend the sequence until a maximal sequence is found. We record this sequence and refer to it as an *initial sequence*. We then remove the leftmost[2] point of the sequence, leaving an ε-regular subsequence (that we need not reconstruct) which we then extend to to the right.

Each time a maximal sequence is found, the leftmost point is removed and we continue extending to the right. When it is no longer possible to continue in this manner (i.e. when the remaining subsequence contains only a single point), we repeat this entire process in the other direction: starting with the current initial sequence, we now remove the rightmost point and extend to the left, etc. We refer to this process of removing and extending as *marching*.

We mark each pair of points that we encounter during the extension process; when choosing new starting pairs, this information helps avoid pairs that have been previously processed. This also serves to prevent *cycling* which could occur if the extending process loops back on itself to yield the same initial sequence (e.g., when the points lie on a large circle).

The points of an *ε-regular sequence* are implicitly ordered by the positions of the corresponding regular points along their containing line. A **remove** operation removes an endpoint from an ε-regular sequence, yielding a shorter ε-regular subsequence. Our algorithm begins with all pairs labeled *unmarked*, and pairs are **mark**ed as they are extended to form maximal ε-regular sequences. An ε-regular sequence is **extend**ed by appending a new point that preserves the ε-regular property (this process is described below in greater detail). A high-level description of our algorithm is given in Figure 3.

[1]Under the Chebyshev metric, the distance between two points (x, y) and (u, v) is given by $\max(|x-u|, |y-v|)$.

[2]One end of the sequence is chosen arbitrarily to be the *left end* of the sequence, while the other end is denoted as the *right end* of the sequence.

Robust Mine Detection (RMD) Algorithm

Input: A pointset $P \subset E^2$
Output: $LIST$ of all maximal ε-regular subsets.

1: $LIST = \emptyset$
2: **For** all *unmarked* pairs $S = \{p_i, p_j\}$
3:     **Call Procedure Initialize** S
4:     **Let** $INIT = S$
       /* *INIT is now the initial sequence* */
5:     **Let** $LIST = LIST \cup \{INIT\}$
       /* *We are now ready to begin extending* */
6:     **Call Procedure March**($right$)
7:     **Unmark** all pairs in $INIT$
8:     **Let** $S = INIT$
9:     **Call Procedure March**($left$)
10:    **Output** S

Procedure Initialize S

1: **While** S can be extended to the left **Do Extend** S to the left
2: **While** S can be extended to the right **Do Extend** S to the right

Procedure March($direction$)

/* *<direction> is either 'right' or 'left'* */
/* *<opposite direction> is 'right' if <direction> is 'left' and vice-versa* */
1: **Remove** the <*opposite direction*>-most point from S
2: **While** $|S| > 1$ and the <*opposite direction*>-most pair of S is *unmarked*
3:     **Mark** the <*opposite direction*>-most pair of S
4:     **If** S is extendable to the <*direction*> **Then**
5:         **Repeat** (**Extend** S to the <*direction*>)
            Until S cannot be extended to the <*direction*>
6:         **Let** $LIST = LIST \cup \{S\}$
7:     **Remove** the <*opposite direction*>-most point from S

Figure 3: An $O(n^{5/2})$-time algorithm (RMD) for the Mine Dection Problem

We can prove the following correctness property of our algorithm (see [7] for details):

Theorem 3.1 *Algorithm* RMD *finds all maximal ε-regular sequences among the input points for all $\varepsilon \geq 0$.*

4 The Algorithm Details

We now expand on some of the subroutines used implicitly in the high-level description of the algorithm. In particular, we describe how we *extend* an ε-regular sequence. Given a sequence $S = (p_i, p_{i+1}, \cdots, p_{j-1}, p_j)$, where the points in S are a subset of the input pointset P, we **extend left** by finding a point $p_{i-1} \in P$ which, when included into S, maintains the ε-regular property of S, and then prefixing the point to yield $S = (p_{i-1}, p_i, p_{i+1}, \cdots, p_{j-1}, p_j)$. Similarly, we **extend right** by finding a point $p_{j+1} \in P$ such that $S = (p_i, p_{i+1}, \cdots p_{j-1}, p_j, p_{j+1})$ is ε-regular, and including that point into P.

We implement the extension operation in two steps - *projection* and *verification*. *Projection* entails determining a set of candidate points for inclusion into a sequence. This method, combined with the minimum interpoint distance restriction discussed earlier, guarantees that at most a single candidate needs to be considered. The *verification* step employs linear programing to verify that S remains ε-regular when augmented with this candidate.

Projection works as follows. Given two points, $p_i, p_j, i < j$ in an ε-regular sequence S, let $k = i - 1$ or $k = j + 1$ correspond to left or right extensions, respectively. The region in which the kth element of S must lie is bounded as follows:

$$p_{k_x} \leq \frac{k-j}{j-i} \cdot (p_{j_x} - p_{i_x} + 2\varepsilon) + p_{j_x} + 2\varepsilon \tag{1}$$

$$p_{k_x} \geq \frac{k-j}{j-i} \cdot (p_{j_x} - p_{i_x} - 2\varepsilon) + p_{j_x} - 2\varepsilon \tag{2}$$

$$p_{k_y} \leq \frac{k-j}{j-i} \cdot (p_{j_y} - p_{i_y} + 2\varepsilon) + p_{j_y} + 2\varepsilon \tag{3}$$

$$p_{k_y} \geq \frac{k-j}{j-i} \cdot (p_{j_y} - p_{i_y} - 2\varepsilon) + p_{j_y} - 2\varepsilon \tag{4}$$

This region, illustrated in Figure 4 with $i = 1, j = 2, k = 3$, is referred to as a *right projection* when $k = j + 1$ and as a *left projection* when $k = i - 1$. We denote the set of points lying in such a region $proj_l(p_i, p_j, j - i)$ or $proj_r(p_i, p_j, j - i)$, for left and right projections respectively. Having determined the region using equations (1 - 4), we may count the points lying inside the region in $O(\log n)$ time [6].

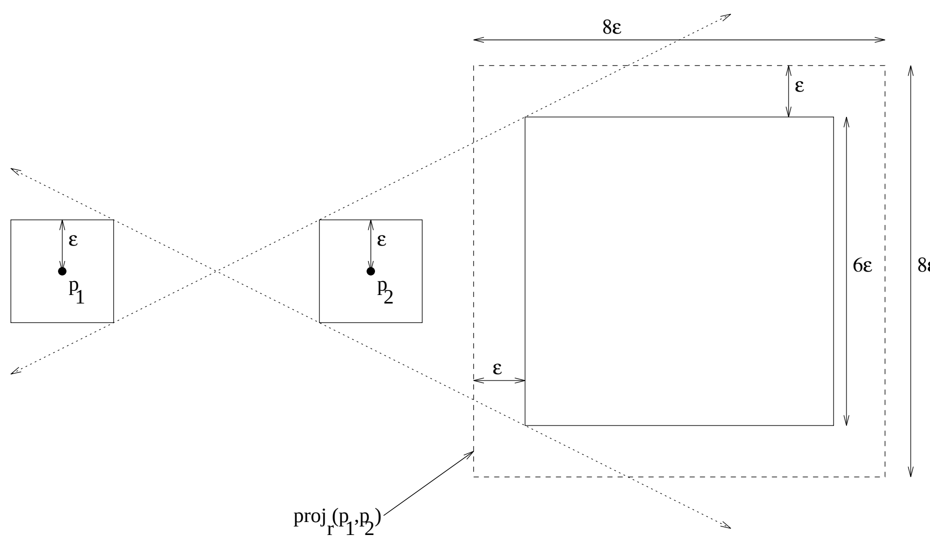

Figure 4: The dashed square indicates the region in which p_3 must lie if (p_1, p_2, p_3) is an ε-regular sequence. The solid square denotes the region in which the corresponding regular point must lie.

While the projection operation utilizes only two of the points in the ε-regular sequence in order to determine the location of a potential new point, the location of a new point to be included is constrained by the locations of *all* the points already in the sequence (not only by a single pair). Thus a verification step which considers all of these constraints is required.

We use linear pogramming to process all of these constraints efficiently. A linear programming instance generally consists of a set of linear inequality constraints and an objective function that must be optimized subject to the constraints. In order to exploit efficient methods for solving low-dimensional linear programs [4], we define the necessary constraints for our linear program to verify ε-regularity by describing a regular sequence by a point and a vector in a manner similar to the point-slope description of a line, using the length of a vector $\vec{v}$ to describe the inter-point distance. That is, a regular sequence R is described by $(r_0, \vec{v})$, $R = \{r_i | r_i = r_0 + \vec{v} \cdot i\}$ for $0 \leq i \leq n - 1$ where $\vec{v}$ is a vector.

In the two-dimensional Chebyshev plane, a sequence is ε-regular if and only if p_i is within distance ε of r_i in both the x and y dimensions. That is, S is ε-regular if and only if its orthogonal projections to the coordinate axes are ε-regular. We may thus use the following constraints in our linear program, where v_x, v_y, r_{0_x} and r_{0_y} are variables:

$$p_{i_x} \geq r_{0_x} + \vec{v}_x \cdot i - \varepsilon \tag{5}$$

$$p_{i_x} \leq r_{0_x} + \vec{v}_x \cdot i + \varepsilon \tag{6}$$

$$p_{i_y} \geq r_{0_y} + \vec{v}_y \cdot i - \varepsilon \tag{7}$$

$$p_{i_y} \leq r_{0_y} + \vec{v}_y \cdot i + \varepsilon \tag{8}$$

Linear programs in fixed dimension (i.e. the number of variables) can be solved in time linear in the number of constraints [4]. In our application, the dimension is fixed at four (the four variables being $r_{0_x}, v_x, r_{0_y}, v_y$). Thus, the time required for the verification step is proportional to the length of the sequence.

The minimum interpoint-distance condition prevents finding more than one candidate in any single projection region: suppose two points may be used to extend an ε-regular sequence in a given direction. Both of these points must lie in the same projection region, which measures 8ε on each side (See Figure 4). Therefore if all interpoint distances are greater than 8ε, we are guaranteed that this will never occur, and this in turn prevents exponential output size. Based on the above techniques, we can prove the following time complexity when interpoint-distances are are greater than 8ε. (see [7] for details):

Theorem 4.1 *The time complexity of algorithm RMD is* $O(n^{5/2})$.

5 Conclusion

We have formulated and addressed the Mine Detection Problem from inexact data. In the unconstrained variant, we have established an exponential lower bound on the output. On the other hand, under reasonable conditions, the problem is solvable within $O(n^{5/2})$ time. Remaining open issues include:

- Giving tight time bounds for solving the problem for minimum interpoint distances of 8ε;
- Generating meaningful polynomially-sized representations of exponentially-sized outputs; and
- Determining the complexity of the Euclidean version (rather than the Chebyshev).

6 Acknowledgments

We are grateful to Dr. Bob Grafton of NSF for his support and encouragement, and to Mike Alexander for his valuable feedback. For more information about our recent work, please see our home page on the World-Wide Web at URL http://www.cs.virginia.edu/~robins/.

References

[1] L. Boxer and R. Miller, *Parallel Algorithms for all Maximal Equally-Spaced Collinear sets and all Maximal Regular Coplanar Lattices*, Pattern Recognition Letters, 14 (1993), pp. 14–20.

[2] L. Guibas, D. Salesin, and J. Stolfi, *Constructing Strongly Convex Approximate Hulls with Inaccurate Primitives*, Algorithmica, 9 (1993), pp. 534–560.

[3] A. B. Kahng and G. Robins, *Optimal Algorithms for Extracting Spatial Regularity in Images*, Pattern Recognition Letters, 12 (1991), pp. 757–764.

[4] N. Megiddo, *Linear Programming in Linear Time When the Dimension is Fixed*, J. Assoc. Comput. Mach., 31 (1984), pp. 114–127.

[5] R. V. Nolan, H. C. Egghart, L. Mittleman, R. L. Brooke, F. L. Foder, and D. L. Gravitte, *MERADCOM Mine Detection Program: 1960 - 1980*, Tech. Rep. Report No. 2294, MERADCOM Belvoir RD & E Center, March 1980.

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

[7] G. Robins and B. L. Robinson, *Pattern Minefield Detection from Inexact Data*, Tech. Rep. CS-95-21, Department of Computer Science, University of Virginia, April 1995.