Available online at www.sciencedirect.com
ScienceDirect
Computational Geometry 39 (2008) 2–13
Computational Geometry
Theory and Applications
www.elsevier.com/locate/comgeo

Algorithms for bivariate zonoid depth ☆

Harish Gopala, Pat Morin *

School of Computer Science, Carleton University,
1125 Colonel By Drive, Ottawa, Canada K1S 5B6

Received 7 January 2005; received in revised form 10 November 2006; accepted 7 May 2007

Available online 3 July 2007

Communicated by P. Bose and T. Fevens

Abstract

Zonoid depth is a definition of data depth proposed by Dyckerhoff et al. [R. Dyckerhoff, G. Koshevoy, K. Mosler, Zonoid data depth: Theory and computation, in: A. Prat (Ed.), COMPSTAT 1996—Proceedings in Computational Statistics, Physica-Verlag, Heidelberg, August 1996, pp. 235–240]. Efficient algorithms for solving several computational problems related to zonoid depth in 2-dimensional (bivariate) data sets are studied. These include algorithms for computing a zonoid depth map, computing a zonoid depth contour, and computing the zonoid depth of a point.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Statistical data depth; Zonoids; Zonotopes

1. Introduction

Statistical data depth (or simply, *depth*) is a means of measuring how deep, or central, a given point p in $\mathbb{R}^d$ is with respect to a given data cloud $\{p_1, \ldots, p_n\}$. This concept provides a center-outward ordering of points in Euclidean space of any dimension and leads to a new non-parametric multivariate statistical analysis in which no distribution assumptions are needed.

Liu et al. [24] describe many different notions of depth such as the half space, the convex hull peeling, the Oja, the simplicial, the majority, and the likelihood depths. Many computational problems associated with such data depth functions are non-trivial to solve efficiently. The design of efficient algorithms for these problems is essential for these depth measures to become useful statistical analysis tools. Computational geometry [32] has been of great help in this respect, and there are many results in the computational geometry literature [1–5,9–12,14,17,19,21–23,26,29,31,33,34,36,37].

Here, we focus on one particular depth measure, *zonoid depth*, introduced by Dyckerhoff et al. [14] and which is the topic of the book by Mosler [30].

☆ This research was partly funded by a grant from the Natural Sciences and Engineering Research Council of Canada and by an Ontario Graduate Scholarship.
* Corresponding author.
E-mail address: morin@scs.carleton.ca (P. Morin).

0925-7721/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2007.05.007

1.1. Definition of zonoid depth and regions

Given a set of points $S = \{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^d$, the convex hull of S is defined as

$$CH(S) = \left\{ \sum_{i=1}^{n} \lambda_i p_i \colon 0 \leqslant \lambda_i \leqslant 1, \ \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

The *k-zonoid* (the zonoid of depth k) is defined as

$$Z_k(S) = \left\{ \sum_{i=1}^{n} \lambda_i p_i \colon 0 \leqslant \lambda_i \leqslant \frac{1}{k}, \ \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

Here, and throughout this article, $1 \leqslant k \leqslant n$ is an integer[1] and we focus on the special case $d = 2$. The *zonoid depth* of a point p with respect to a set S is defined as the maximum value k for which p is contained in $Z_k(S)$.

Since a zonoid is defined by a finite set of linear constraints, it forms a convex polygon. Furthermore, for $k_1 > k_2$, $Z_{k_1}(S) \subseteq Z_{k_2}(S)$, hence $Z_1(S), \ldots, Z_n(S)$ form a sequence of nested convex polygons. The n-zonoid $Z_n(S)$ contains a single point, the mean of S. For other properties of zonoids, see Dyckerhoff et al. [14] and Mosler [30].

1.2. Summary of results and related work

In this paper, we give the following algorithmic results for zonoid depth:

(1) *Computing a contour*: an $O(n \log n + nk^{1/3})$ expected time algorithm to compute the convex polygon bounding $Z_k(S)$.
(2) *Computing a depth map*: an $O(n^2)$ time algorithm to compute $Z_1(S), \ldots, Z_n(S)$.
(3) *Testing the depth*: an $O(n)$ time algorithm to test whether a point p is contained in $Z_k(S)$.
(4) *Computing the depth*: an $O(n)$ expected time algorithm to compute the zonoid depth of a point p.

Algorithms 2, 3 and 4 are optimal. Improving Algorithm 1 would require an improvement to current bounds on the maximum number of k-sets of a planar point set.

Dyckerhoff et al. [14] give an algorithm to compute the depth of a point in a data cloud of fixed dimension d by solving a linear program in the variables $\lambda_1, \ldots, \lambda_n$. To obtain an efficient algorithm, they make use of the fact that most of the constraints on the λ_i's are independent of S. However, the worst-case running time of their algorithm is unclear.

The results in this paper are obtained by exploiting an observation relating k-zonoids and k-sets. This observation has also been used by Bern and Eppstein [6] in the context of support vector machines in machine learning. (They refer to zonoids as *reduced convex hulls*.) In particular, Bern and Eppstein show that, given two sets S_1 and S_2 in $\mathbb{R}^d$, computing the smallest value k such that $Z_k(S_1) \cap Z_k(S_2)$ is non-empty can be solved using a number of arithmetic operations that is linear in n and polynomial in d, L and $\log n$. Here, L is the number of bits used to represent coordinates of the input points. Their algorithm uses Khachiyan's ellipsoid method [18] for linear programming. In the conclusions of their paper, Bern and Eppstein suggest that it may be possible to solve low-dimensional versions of their problem using generalized linear programming (GLP) [27]. However, we have found that this is not so easy and that, even in 2 dimensions, we require some techniques that go beyond those of GLP.

The remainder of this article is organized as follows: In Section 2, we describe the relationship between zonoids, k-sets and k-levels. In Section 3, we develop algorithms for zonoid depth problems. In Section 4, we summarize our results and list some open problems pertaining to zonoid depth.

[1] In the zonoid depth literature, the value k may be any real number between 1 and n. Although the algorithms in this paper are stated for k being an integer, they extend easily to the case of real-valued k.

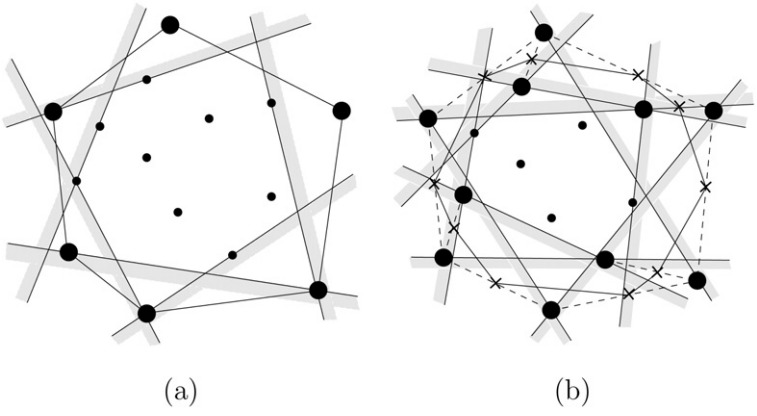


Fig. 1. A point set S and (a) 1-sets on S and the 1-zonoid and (b) 2-sets on S and the 2-zonoid.

2. Zonoids, k-sets and k-levels

In this section we review some background material on duality and discuss the relationships between k-zonoids, k-sets, and k-levels.

2.1. Correspondence between zonoids and k-sets

A point set is said to be in *general position* if no two points of the set lie on a vertical line and no three points of the set are collinear. Here, and throughout the article, point sets are always assumed to be in general position.

Given a set S of n points in general position and an integer $0 \leqslant k \leqslant n-2$, a set $S' \subseteq S$ is called a k-set of S if $|S'| = k$ and there exists a closed halfplane h such that $h \cap S = S'$. I.e., S' has k points in it and these can be separated from the remaining $n-k$ points of S with a straight line. The notion of k-sets was introduced by Erdös et al. [16] and it is a long standing open problem to determine the maximum number of k-sets in a set of n points. Currently, the best bound of $O(nk^{1/3})$ is due to Dey [13].

Consider a set S of n points. If we construct all possible 1-sets on S, we obtain the vertices of $CH(S)$, represented by the thick points in Fig. 1(a). By taking the convex hull of all such 1-set points, we obtain $Z_1(S)$, i.e., $Z_1(S) = CH(S)$, as the definition implies.

Now, on the same set S, construct all possible 2-sets. In each 2-set, take the mean (represented, in Fig. 1(b), by the **X** on the dotted line segment joining 2 points in each 2-set) of the pair of points. The following lemma shows that by taking the convex hull of the means from all 2-sets of S, we obtain the 2-zonoid of S, i.e., $Z_2(S)$ as in Fig. 1(b). In a similar fashion, zonoids up to depth n can be constructed. Since $Z_n(S)$ is the mean of the points in S, it is a unique point and is the center of gravity of the points in S.

Lemma 1. *Given a set S of n points in $\mathbb{R}^2$ and an integer $1 \leqslant k \leqslant n$, there is a bijection between the vertices of $Z_k(S)$ and the k-sets of S.*

Proof. Consider the mapping f that takes a k-set $S' \subseteq S$ onto a vertex of $Z_k(S)$ using the equation $f(S') = \frac{1}{k}\sum_{p \in S'} p$. We will show that f is a bijection between the k-sets of S and the vertices of $Z_k(S)$.

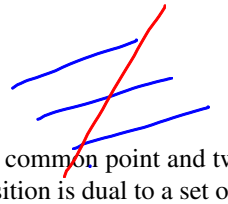
To see that f is one-to-one, consider two k-sets S_1 and S_2. We need to show that the points $p_1 = \frac{1}{k}\sum_{p \in S_1} p$ and $p_2 = \frac{1}{k}\sum_{p \in S_2} p$ are distinct. The set S_1 is the intersection of S with a halfplane h bounded by a line ℓ having inner normal d. The numbers $S_1 \cdot d = \{\{p \cdot d\colon\ p \in S'\}\}$ are the k largest values in the multiset $S \cdot d = \{\{p \cdot d\colon\ p \in S\}\}$. It follows that $p_1 \cdot d > p_2 \cdot d$, so $p_1 \neq p_2$, as required.

To see that f is onto, we observe that any vertex v of $Z_k(S)$ is extreme in some direction d. In fact, $v = \frac{1}{k}\sum_{x \in S'} x$, where S' is a subset of the extreme-most k points of S in direction d. But then S' is a k-set of S since it can be separated from S by a line ℓ perpendicular to d. □

2.2. Review of duality

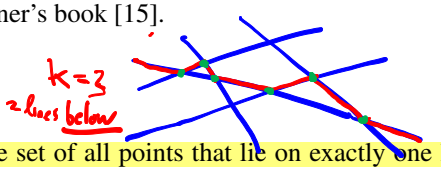
Let $p = (p_1, p_2)$ denote a point in the plane. The dual of p, denoted by p^*, is the line $p^* = \{(x, y)\colon\ y = p_1 x - p_2\}$. The dual of the line $l = \{(x, y)\colon\ y = ax + b\}$ is the point $l^* = (a, -b)$. We say that the duality transform maps objects from the *primal* plane to the *dual* plane. This transform has the following properties:

- It is *incidence preserving*: $p \in l$ if and only if $l^* \in p^*$.
- It is *order preserving*: p lies above l if and only if l^* lies above p^*.



A set of lines is said to be in general position if no three of the lines pass through a common point and two of the lines are parallel. It follows from the above properties that a set of points in general position is dual to a set of lines in general position. For further details on duality see, e.g., Edelsbrunner's book [15].

2.3. Correspondence between zonoids and levels



The k-level of a set L if n lines is defined as the closure of the set of all points that lie on exactly one line of L and strictly above $k-1$ lines of L. Note that the reflex vertices of the k-level are points that are on two lines and that have $k-2$ lines below them. If the lines in L are dual to a set P of points then each reflex vertex p on the k-level of L is the dual of a line p^* that bounds (from above) a closed halfplane containing k points (a k-set) of P. Similarly, a convex vertex on the $(n-k)$-level is dual to a line bounding (from below) a closed halfplace containing a k-set of P.

We describe the k-zonoid in both the primal and the dual settings and show the relationship between them. The left part of Fig. 2 represents the primal and the right part, the dual. The upper (lower) convex hull of points in the primal corresponds to the upper (lower) envelope of the dual lines in the dual. In the primal, we construct a k-zonoid, for some k. In the dual, this is the shaded region. The upper and lower boundaries of the shaded region are also convex, because the corresponding boundaries of the k-zonoid in the primal are convex.

The upper half of the k-zonoid in the primal can be constructed (inefficiently) by finding all possible k-sets defined by halfplanes bounded from below, taking the mean of the k points in each k-set and taking the convex hull of the resulting point set. In the dual, this corresponds to constructing the k-level and then, for each reflex vertex drawing a closed downwards vertical ray and computing the mean y-coordinate of the k lines that intersect this ray. The resulting points are then connected by increasing x-coordinates to obtain the boundary the dual k-zonoid (the shaded region in Fig. 2). The lower half of the k-zonoid can be constructed in a similar manner starting the with $(n-k)$-level.

Although the number of k-sets and the number of vertices of the k-level are different, their worst-case complexities are within a constant factor of each other [15]. Dey [13] proves an $O(nk^{1/3})$ upper bound on the complexity of planar k-levels, which is also an upper bound for the number of planar k-sets. Since we showed that there is a bijection between the k-sets of a point set S and the vertices of $Z_k(S)$ in Lemma 1, Dey's result implies an $O(nk^{1/3})$ upper bound on the number of vertices of a k-zonoid in $\mathbb{R}^2$.

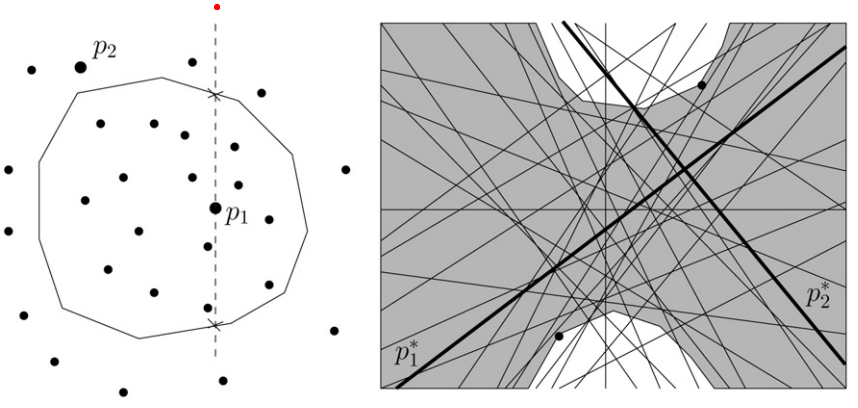


Fig. 2. A k-zonoid in primal and dual.

Sharir et al. [35] show that the number of k-sets in a set of n points in $\mathbb{R}^3$ is $O(nk^{3/2})$. This implies that, in $\mathbb{R}^3$, the k-zonoid of an n point set has size $O(nk^{3/2})$.

3. Algorithms for zonoids in 2 dimensions

In this section we present algorithms for bivariate zonoid depth. The first few algorithms follow easily from existing results on k-sets and arrangements. The later algorithms are somewhat more involved.

3.1. Computing a depth contour

Chan [8] describes an algorithm for computing the k-level in a set of n lines that runs in $O(n \log n + nk^{1/3})$ expected time. Using this algorithm, the k-level and the $(n-k)$-level can be constructed in $O(n \log n + nk^{1/3})$ expected time. This algorithm is easily augmented to output the k-zonoid.

Theorem 1. *Given a set S of n points in $\mathbb{R}^2$ and an integer $1 \leqslant k \leqslant n$, the k-zonoid $Z_k(S)$ can be computed in $O(n \log n + nk^{1/3})$ expected time.*

3.2. Computing a depth map

Applying Theorem 1 n times for k from 1 to n gives us an algorithm to compute a depth map in $O(n^2 \log n + n^2 k^{1/3})$ expected time. But the relationship between k-zonoids, k-levels and $(n-k)$-levels allows us to compute $Z_1(S), \ldots, Z_n(S)$ in $O(n^2)$ time by computing the arrangement of the dual lines [15].

Theorem 2. *Given a set S of n points in $\mathbb{R}^2$, $Z_1(S), \ldots, Z_n(S)$ (i.e. the depth map) can be computed in $O(n^2)$ time.*

Once we have computed the $Z_1(S), \ldots, Z_n(S)$, we can preprocess them for point location using Kirkpatrick's planar point location algorithm [20] so that we can determine the zonoid depth of any point in $O(\log n)$ time.

Theorem 3. *Given a set S of n points in $\mathbb{R}^2$, after preprocessing requiring $O(n^2)$ time and space the zonoid depth of any query point p can be computed in $O(\log n)$ time.*

3.3. Testing if a zonoid contains a point

In this section, we study the following decision problem: Given a set S of n points in general position, a query point p and an integer $1 \leqslant k \leqslant n$, determine whether p is contained in $Z_k(S)$.

Consider again Fig. 2. In the primal, if the point p_1 were to be moved upwards along a vertical line passing through p_1, then the line p_1^* also moves upwards in the dual. When p_1 hits the k-zonoid boundary, p_1^* becomes tangent to the boundary of the dual of the k-zonoid. This leads to the following idea: In the primal, first determine the vertices of the zonoid with smallest and largest x-coordinate. If the point p_1 is not in the vertical strip between these two points then p_1 is not the in the zonoid.

Otherwise, draw a vertical line through the point p_1. This line intersects the boundary of the k-zonoid in 2 points. Finding these intersection points is equivalent to finding the points at which a vertical translation of line p_1^* becomes tangent to the boundaries of the dual of the k-zonoid. Once they are found, it can be easily determined whether p_1 is inside or outside the k-zonoid by comparing the y-coordinates of the intersection points with that of p_1. Hereafter, we concentrate on finding that vertex on the upper boundary of the dual of the k-zonoid at which p_1^* is tangent. Such a vertex on the lower boundary can be found in a symmetric manner.

The algorithm that we use is inspired by the planar ham-sandwich algorithm of Lo et al. [25]. In their algorithm, Lo et al. are searching for a particular vertex on the median level of the arrangement of lines in the dual. In the primal, this vertex corresponds to a line passing through 2 points and bisecting this set. In our problem, we are searching for a particular vertex on the k-level that defines the point on the upper boundary of the dual of the k-zonoid at which a vertical translation of p_1^* is tangent.

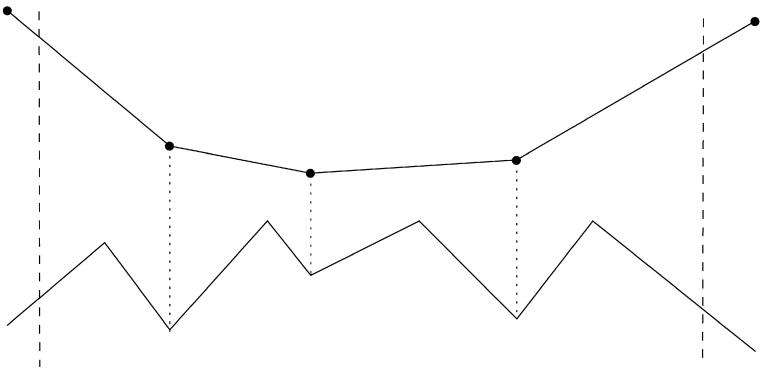


Fig. 3. Vertical strip V showing the k-level and corresponding upper boundary of the dual of the k-zonoid.

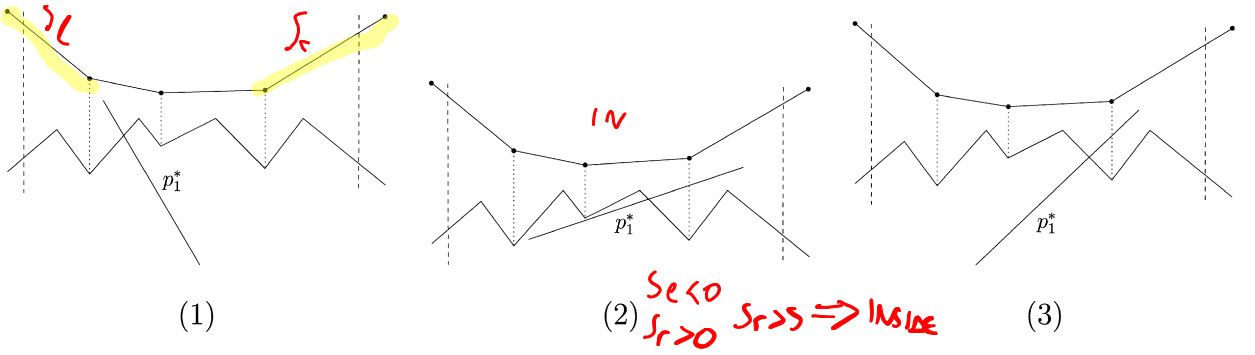


Fig. 4. The vertical translation of p_1^* is tangent to the upper boundary of the dual of the k-zonoid (1) to the left of V, (2) inside of V and (3) to the right of V.

In the following, we describe and analyze our algorithm in detail even though the algorithm and its analysis are more or less the same as the algorithm of Lo et al. This is because we will be modifying this algorithm in Section 3.4 to solve a weighted version of problem. Describing and analyzing the modified algorithm requires an understanding of the original algorithm and its analysis.

Consider an open vertical strip V in the dual, as in Fig. 3, showing the k-level and the corresponding convex upper boundary of the dual of the k-zonoid. Refer to Fig. 4. Our algorithm requires a method to determine whether a vertical translation of p_1^* becomes tangent to the upper boundary of the dual k-zonoid (1) to the left of the strip V, (2) in the strip V, or (3) to the right of the strip V. To determine this, we compute the slope s_ℓ of the dual k-zonoid edge that intersects the left boundary of V and the slope s_r of the dual k-zonoid edge that intersects the right boundary of V. Both these computations can be done in $O(n)$ time using an $O(n)$ time selection algorithm to select the intersection of the k-level with each of the two vertical lines bounding V. If s is the slope of p^* then either

(1) $s \leqslant s_\ell < s_r$ in which case a vertical translation of p_1^* is tangent to the dual k-zonoid at some point to the left of V,
(2) $s_\ell < s < s_r$ in which case a vertical translation of p_1^* is tangent to the dual k-zonoid at some point in V, or
(3) $s_\ell < s_r \leqslant s$ in which case a vertical translation of p_1^* is tangent to the dual k-zonoid at some point to the right of V.

The above method is used in conjunction with the following lemma (that appears in Ref. [25]) for subdividing an arrangement into vertical strips:

Subdivision Lemma 1. *Let L be a set of n lines in the plane in general position, let $\alpha < 1$ be a prescribed positive constant and let V be a vertical strip. In $O(n)$ time, V can be partitioned into vertical strips $V_1, V_2, \ldots, V_C$ (where $C = C(\alpha) \leqslant 2/\alpha$ is a function that depends only on α), such that each V_i contains at most αN of the $N = \binom{n}{2}$ intersection points between pairs of lines in L.*

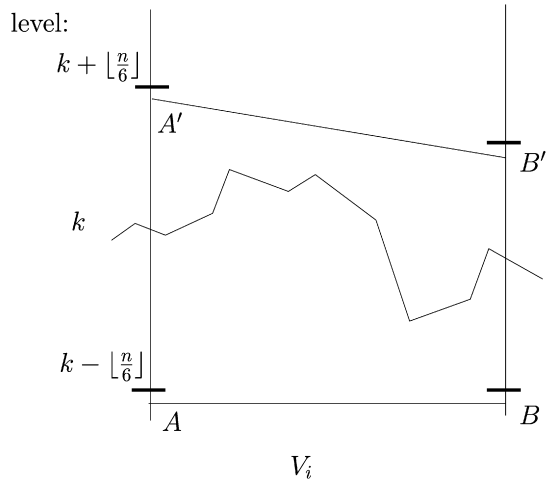


Fig. 5. Figure showing how we build the trapezoid $ABB'A'$.

In our problem, we apply the Subdivision Lemma in the dual. Let our strip $V = \mathbb{R}^2$. We want to subdivide V into substrips so that each substrip contains at most $\binom{n}{2}/20$ of the total number of intersections in V. The reason for this choice will become clear shortly. Setting $\alpha = 1/20$ gives at most $C(1/20) \leqslant 40$ strips $V_1, V_2, \ldots, V_{40}$.

We now have 40 open vertical strips and we can determine in $O(n)$ time, using the method described above, a strip that V_i contains the upper boundary vertex at which a line parallel to p_1^* is tangent.

In V_i, we build a trapezoid $T = ABB'A'$, as shown in Fig. 5. The points A on the left vertical line and B on the right vertical line lie just below the $k - \lfloor \frac{n}{6} \rfloor$ level of the lines in the dual and the points A' and B' lie just below the $k + \lfloor \frac{n}{6} \rfloor$ level of the lines in the dual (if $k - \lfloor \frac{n}{6} \rfloor \leqslant 0$, then A and B are chosen below all lines, and similarly for the top side).

Lemma 2. *The top and bottom sides of the trapezoid T intersects at most $n/3$ dual lines each.*

Proof. Let u be the number of lines intersecting the side AB of T *upwards*, i.e., these lines have A above and B below them. Let d be the number of lines intersecting AB of T *downwards*. Since A and B are each on the $k - \lfloor \frac{n}{6} \rfloor$ level, we have $u = d$. But every *upwards* line intersects every *downwards* line in the strip V_i, hence V_i contains at least $ud = u^2$ intersections.

Each strip V_i contains at most $\frac{1}{20}\binom{n}{2}$ intersections. Therefore, $u^2 \leqslant \binom{n}{2}/20$, which implies that $u \leqslant n/\sqrt{40} \leqslant n/6$. Thus, $u + d \leqslant n/3$ is the number of lines intersecting side AB of trapezoid T. A symmetric argument shows that the number of lines intersecting $A'B'$ is also at most $n/3$. □

By the choice A, B, B' and A', each of the two vertical sides of the trapezoid T are also intersected by at most $\frac{n}{3}$ lines. Thus, each side of T is intersected by $\leqslant \frac{n}{3}$ lines, for a total of $\frac{4n}{3}$ intersections altogether. Each line that intersects the interior of T contributes 2 to this sum, so at most $\frac{2n}{3}$ lines intersect the interior of T and at least $\frac{n}{3}$ lines do not intersect the interior of T.

Lemma 3. *Within the vertical strip V, the k-level is completely contained in T.*

Proof. Suppose that the k-level goes below the bottom side AB of T. Then some point C on the segment AB has level greater than k. Since both A and B have level $k - \lfloor \frac{n}{6} \rfloor$, each of the segments AC and BC must be intersected by more than $\frac{n}{6}$ lines. But this contradicts Lemma 2. Similarly, the k-level can not be above the top side $A'B'$ of T and must therefore be contained in T. □

Based on Lemmata 2 and 3, the $\frac{n}{3}$ lines that pass outside trapezoid T can be safely discarded since the intersection of the k-level with the strip V_i is completely contained in T. However, when we discard the lines that lie above T, we remember their mean, since this will be needed to compute slopes in recursive calls to the algorithm.

After we build the trapezoid T and discard $\frac{n}{3}$ of the lines passing outside T, we reconstruct open vertical strips inside this trapezoid for the remaining $\frac{2n}{3}$ lines and get a new trapezoid and search within it. Thus, each iteration (or recursive invocation) runs in time linear in the number of lines and discards a constant fraction of the lines. Hence the algorithm runs in $O(n)$ time. This yields the following theorem.

Theorem 4. *Given a set S of n points in $\mathbb{R}^2$, a query point p and an integer $1 \leqslant k \leqslant n$, we can determine in $O(n)$ time whether or not p lies inside or outside the k-zonoid $Z_k(S)$. More generally, we can compute the intersection of $Z_k(S)$ with any line in $O(n)$ time.*

3.4. Computing the zonoid depth of a point

Theorem 4 offers a means of solving the *decision problem*: Is $p \in Z_k(S)$? In this section we solve the optimization problem: What is the zonoid depth of p? I.e., what is the maximum value k such that $p \in Z_k(S)$? To do this, we use a general technique due to Chan [7] for converting decision algorithms into optimization algorithms. Let $depth(p, S)$ denote the zonoid depth of p with respect to S. Roughly stated, Chan's algorithm requires

(1) (Decision Algorithm) a decision algorithm to decide whether $depth(p, S)$ is at least k and
(2) (Decomposition into Subproblems) an algorithm to express $depth(p, S)$ as $\max\{depth(p, S_i)\colon 1 \leqslant i \leqslant r\}$ where r is a constant and each $|S_i| \leqslant \alpha|S|$ for some constant $\alpha < 1$.

If these requirements are met, then Chan's technique allows us to solve the optimization problem efficiently.[2] Initially, it seems that Theorem 4 satisfies Requirement 1 above. However, to satisfy Requirement 2 we need to define a slightly more general problem on weighted zonoids. This, in turn, means that we need a slightly more general algorithm to satisfy Requirement 1.

3.4.1. Weighted zonoids

Let $S = \{p_1, p_2, \ldots, p_m\}$ be a set of m points in general position and let each point p_i have an associated positive integer weight $w(p_i) = w_i$. Let $n = \sum_{i=1}^{m} w_i$. The *w-weighted k-zonoid of S* is defined as

$$Z_k(S, w) = \left\{ \sum_{i=1}^{m} \lambda_i w_i p_i \;\middle|\; 0 \leqslant \lambda_i \leqslant \frac{1}{k}, \sum_{i=1}^{n} w_i \lambda_i = 1 \right\}.$$

Note that the original definition of zonoid depth is a special case of this definition in which $n = m$ and $w_i = 1$ for all $1 \leqslant i \leqslant m$. Also observe that, since the w_i are positive integers, any problem on weighted zonoids can be converted into a similar unweighted zonoid problem by replacing each point p_i by w_i infinitesimally close points. Thus, for example, an algorithm similar to that of Theorem 4 could be applied to compute the intersection of any line with a weighted zonoid in $O(n)$ time. However, this observation is insufficient for our purposes, and we need an algorithm whose running time depends more on the number of points m than the total weight n of those points.

Lemma 4. *Let S, n, w and m be defined as above. Then, the intersection of any line with $Z_k(S, w)$ can be computed in $O(m \log(n/m) + m)$ time.*

Proof. We apply a slight modification of the algorithm given by Theorem 4. In this modification, the algorithm works on the *weighted k-level*. (The weight of a point p is the sum of the weights of all lines that pass through or above p. The weighted k-level is the lower envelope of all points having weight at least k.) The modified algorithm is almost identical to the previous algorithm except that, where the previous algorithm uses a linear-time selection algorithm to compute the intersection of the k-level with a vertical line, the modified algorithm makes use of a linear-time weighted selection algorithm. Also, where the previous algorithm uses the Subdivision Lemma, the modified algorithm uses a weighted Subdivision Lemma which guarantees that each open strip contains at most $\binom{n}{2}/20$ *weighted intersections*.

[2] There is a more precise statement of Chan's very powerful optimization technique, but our application of the technique does not fit the usual requirements of this statement, so we omit it here. The interested reader is referred to Chan's original paper [7].

(The weight of an intersection between line p_i^* and p_j^* is $w_i \times w_j$.) Otherwise, the algorithm is identical to the previous algorithm. At each round, the algorithm runs in $O(m)$ time, where m is the number of lines at the beginning of the round and reduces the total weight n of those lines (but not necessarily the number of lines m) by a factor of $2/3$. Correctness of the algorithm follows from exactly the same argument as the original algorithm.

To analyze the running time of the modified algorithm we observe that the ith iteration of the algorithm takes $O(m_i)$ time, where m_i is the number of lines at the beginning of the ith iteration. Furthermore, $m_i \leqslant n_i$, where n_i is the total weight of those lines. Finally, the value of n_i decreases by a factor of $2/3$ during every iteration. These observations imply that the overall running time of the algorithm is bounded by

$$T_{n,m} \leqslant \sum_{i=0}^{\infty} O(m_i) \leqslant \sum_{i=0}^{\infty} O\big(\min\{(2/3)^i n, m\}\big) = O\big(m \log(n/m)\big) + O(m),$$

as required. □

3.4.2. The decomposition into subproblems

The algorithm for weighted zonoids from the previous section will play the role of the decision algorithm in our application of Chan's optimization technique. In this section, we describe the decomposition into subproblems required to apply Chan's technique.

Assume S, w and p are the inputs to our problem. That is, we want to compute the w-weighted zonoid depth of p with respect to S. We can decompose this problem into 4 subproblems S_1, S_2, S_3, and S_4 as follows: In $O(m)$ time, we select two lines that partition the set S into 4 quadrants Q_1, Q_2, Q_3, and Q_4 that each contain roughly $m/4$ points, using Megiddo's algorithm [28]. Subproblem S_1 contains 3 consecutive quadrants, say Q_1, Q_2, Q_3, and a single point which is the weighted average of all the points in Q_4 and whose weight is the sum of the weights of all points in Q_4. That is we create a new point p_ℓ with,

$$w_\ell = \sum_{p_i \in Q_4} w_i$$

and

$$p_\ell = \frac{1}{w_\ell} \sum_{p_i \in Q_4} w_i p_i.$$

Thus, S_1 contains $\frac{3m}{4} + 1$ points. We define the sets S_2, S_3, and S_4 in a similar manner. We define $\mathit{depth}(p, S_i)$ as the weighted zonoid depth of point p in subproblem S_i. It follows immediately from the definition of weighted zonoids that this decomposition produces smaller zonoids, i.e. $Z_k(S_i, w) \subseteq Z_k(S, w)$ and does not change the center, i.e., $Z_n(S_i) = Z_n(S)$. The following lemma shows that this decomposition satisfies the second requirement of Chan's optimization technique.

Lemma 5. *Let S and S_1, S_2, S_3, and S_4 be defined as above. Then*

$$\mathit{depth}(p, S) = \max\big\{\mathit{depth}(p, S_i) \mid 1 \leqslant i \leqslant 4\big\}.$$

Proof. That $\mathit{depth}(p, S_i) \leqslant \mathit{depth}(p, S)$ for all $1 \leqslant i \leqslant 4$ follows from the observation that $Z_k(S_i) \subseteq Z_k(S)$. Thus, all that needs to be shown is that there exists some i, $1 \leqslant i \leqslant 4$, with $\mathit{depth}(p, S_i) = \mathit{depth}(p, S)$.

Suppose $\mathit{depth}(p, S) = k$ and partition $Z_k(S)$ into triangles by drawing segments joining $Z_n(S)$ to each of the vertices of $Z_k(S)$, as shown in Fig. 6. The point p lies in one of these triangles, say with vertices $Z_n(S)$, a and b. The points a and b correspond to two k-sets that have $k-1$ points in common. Indeed, there are two infinitesimally close lines l_a and l_b such that l_a defines the k-set for a and l_b defines the k-set for b. Since l_a and l_b are infinitesimally close, they intersect at most three of the open quadrants $Q_1, \ldots, Q_4$. Wlog suppose they miss Q_4. Then it is not hard to see that $Z_k(S_1, w)$ has a and b as vertices. Furthermore, $Z_k(S_1, w)$ contains $Z_n(S)$ and is convex, so it contains p. Therefore $\mathit{depth}(p, S_1) \geqslant k = \mathit{depth}(p, S)$ as required. □

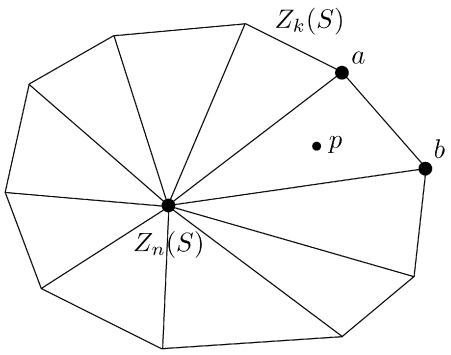


Fig. 6. Partitioning the $Z_k(S)$ zonoid into triangles.

3.4.3. Analysis of Chan's technique

Next, we analyze the cost of applying Chan's optimization technique with the decision algorithm from Section 3.4.1 and the decomposition algorithm from Section 3.4.2. This is not quite a standard application of Chan's technique because the running time of our decision algorithm is a function of both n and m, and our decomposition algorithm only decreases the value of m (and not the value of n) in each subproblem.

If we redo Chan's analysis, we find that the expected number of decision problems we solve at the ith level of recursion is r^i and these decision problems each have total weight n distributed among $m_i = \alpha^i n$ points. Here, $r = \ell \ln 4 + 1$, $\alpha = (3/4)^\ell$ and ℓ is an integer parameter that is under our control. Therefore, the total expected cost of the algorithm is bounded by the summation

$$T_n = \sum_{i=0}^{\infty} \mathrm{O}\big(r^i m_i \log(n/m_i)\big) = n \sum_{i=0}^{\infty} \mathrm{O}\big(r^i \alpha^i \log\big(1/\alpha^i\big)\big) = n \sum_{i=0}^{\infty} \mathrm{O}\big(r^i \alpha^i i\big),$$

which solves to $\mathrm{O}(n)$ provided that $r\alpha < 1$. The condition $r\alpha < 1$ is easily ensured by choosing a sufficiently large value of ℓ. This completes the proof of our final theorem:

Theorem 5. *Given a set S of n points in $\mathbb{R}^2$ and a query point p, we can find the largest integer k for which p lies inside $Z_k(S)$ (i.e., the zonoid depth of p) in $\mathrm{O}(n)$ expected time.*

4. Conclusions and open problems

We have given algorithms for solving several computational problems related to zonoid depth for 2-dimensional (bivariate) data. In particular, we have given

(1) an $\mathrm{O}(n \log n + nk^{1/3})$ algorithm to compute $Z_k(S)$, i.e. the zonoid depth contour of depth k,
(2) an $\mathrm{O}(n^2)$ algorithm to compute $Z_1(S), \ldots, Z_n(S)$, i.e. the zonoid depth map,
(3) a linear time algorithm to test whether a zonoid $Z_k(S)$ contains a point p, and
(4) a linear time algorithm to compute the zonoid depth of a point p.

Results 2, 3 and 4 are optimal. An improvement on Algorithm 1 would require a breakthrough on the (30 year old) planar k-set problem.

This article only deals with zonoid depth problems in 2 dimensions and, besides the work by Dyckerhoff et al. [14] and Bern and Eppstein [6], no work has been done regarding zonoid depth problems in dimensions 3 and higher. In particular, for small (constant) dimensions it is a challenge to find algorithms for computing zonoid depth that do not rely on linear programming in high (a function of n) dimensions.

References

[1] G. Aloupis, On computing geometric estimators of location, Master's thesis, School of Computer Science, McGill University, Montreal, Canada, March 2001.

[2] G. Aloupis, S. Langerman, M. Soss, G. Toussaint, Algorithms for bivariate medians and a Fermat–Torricelli problem for lines, Computational Geometry: Theory and Applications 26 (1) (August 2003) 69–79.

[3] G. Aloupis, E. McLeish, A lower bound for computing Oja depth, Manuscript, School of Computer Science, McGill University, Montreal, Canada, 2004.

[4] G. Aloupis, M. Soss, G. Toussaint, On the computation of the bivariate median and a Fermat–Torricelli problem, Technical Report SOCS-01.2, School of Computer Science, McGill University, Montreal, Canada, February 2001.

[5] N. Amenta, M. Bern, D. Eppstein, S. Teng, Regression depth and center points, Discrete and Computational Geometry 23 (3) (March 2000) 305–323.

[6] M.W. Bern, D. Eppstein, Optimization over zonotopes and training support vector machines, in: Workshop on Algorithms and Data Structures, 2001, pp. 111–121.

[7] T. Chan, Geometric applications of a randomized optimization technique, Discrete and Computational Geometry 22 (4) (December 1999) 547–567.

[8] T. Chan, Remarks on k-level algorithms in the plane, Manuscript, Department of Computer Science, University of Waterloo, Waterloo, Canada, July 7 1999.

[9] T. Chan, An optimal randomized algorithm for maximum Tukey depth, in: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004), SIAM, January 2004, pp. 430–436.

[10] B. Chazelle, On the convex layers of a planar set, IEEE Transactions on Information Theory IT-31 (4) (July 1985) 509–517.

[11] A. Cheng, M. Ouyang, On algorithms for simplicial depth, Technical Report dcs-tr-368, Department of Computer Science, Rutgers University, 1998.

[12] R. Cole, M. Sharir, C.K. Yap, On k-hulls and related problems, SIAM Journal on Computing 16 (1) (February 1987) 61–77.

[13] T.K. Dey, Improved bounds on planar k-sets and k-levels, in: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS 1997), IEEE Computer Society, October 1997, pp. 156–161.

[14] R. Dyckerhoff, G. Koshevoy, K. Mosler, Zonoid data depth: Theory and computation, in: A. Prat (Ed.), COMPSTAT 1996—Proceedings in Computational Statistics, Physica-Verlag, Heidelberg, August 1996, pp. 235–240.

[15] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, New York, 1987.

[16] P. Erdös, L. Lovász, A. Simmons, E.G. Straus, Dissection graphs of planar point sets, in: A Survey of Combinatorial Theory, North-Holland, 1973, pp. 139–149.

[17] S. Jadhav, A. Mukhopadhyay, Computing a centerpoint of a finite planar set of points in linear time, Discrete and Computational Geometry 12 (1994) 291–312.

[18] L.G. Khachiyan, A polynomial time algorithm for linear programming, Soviet Mathematics Doklady 20 (1979) 1092–1096.

[19] S. Khuller, J.S.B. Mitchell, On a triangle counting problem, Information Processing Letters 33 (6) (February 1990) 319–321.

[20] D.G. Kirkpatrick, Optimal search in planar subdivisions, SIAM Journal on Computing 12 (1) (February 1983) 28–35.

[21] S. Langerman, Algorithms and data structures in computational geometry, PhD thesis, Department of Computer Science, Rutgers University, New Brusnwick, May 2001.

[22] S. Langerman, W. Steiger, The complexity of hyperplane depth in the plane, Discrete and Computational Geometry 30 (2) (August 2003) 299–309.

[23] S. Langerman, W. Steiger, Optimization in arrangements, in: H. Alt, M. Habib (Eds.), Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2003), in: Lecture Notes in Computer Science, Springer-Verlag, February 2003, pp. 50–61.

[24] R. Liu, J.M. Parelius, K. Singh, Multivariate analysis by data depth: Descriptive statistics, graphics and inference, The Annals of Statistics 27 (3) (June 1999) 783–858.

[25] C.-Y. Lo, J. Matousek, W. Steiger, Algorithms for ham-sandwich cuts, Discrete and Computational Geometry 11 (1994) 433–452.

[26] J. Matoušek, Computing the center of planar point sets, in: J.E. Goodman, R. Pollack, W. Steiger (Eds.), Computational Geometry: Papers from the DIMACS special year, vol. 6, American Mathematical Society, Providence, RI, 1991, pp. 221–230.

[27] J. Matoušek, M. Sharir, E. Welzl, A subexponential bound for linear programming, Algorithmica 16 (1996) 498–516.

[28] N. Megiddo, Partitioning with two lines in the plane, Journal of Algorithms 6 (3) (September 1985) 430–433.

[29] K. Miller, S. Ramaswami, P. Rousseeuw, T. Sellarès, D. Souvaine, I. Streinu, A. Struyf, Fast implementation of depth contours using topological sweep, in: Proceedings of the 12th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 2001), SIAM, January 2001, pp. 690–699.

[30] K. Mosler, Multivariate Dispersion, Central Regions and Depth. The Lift Zonoid Approach, Lecture Notes in Statistics, vol. 165, Springer-Verlag, New York, 2002.

[31] A. Niinimaa, H. Oja, J. Nyblom, Statistical algorithms: Algorithm AS 277: The Oja bivariate median, Journal of Applied Statistics 41 (3) (1992) 611–617.

[32] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, Springer-Verlag, 1985.

[33] P.J. Rousseeuw, I. Ruts, Constructing the bivariate Tukey median, Statistica Sinica 8 (3) (July 1998) 827–839.

[34] M. Shamos, Geometry and statistics: Problems at the interface, in: Proceedings of the Symposium on Algorithms and Complexity, Carnegie-Mellon University, Pittsburgh, PA, Academic Press, New York, 1976, pp. 251–280.

[35] M. Sharir, S. Smorodinsky, G. Tardos, An improved bound for k-sets in three dimensions, in: Proceedings of the 16th Annual Symposium on Computational Geometry (SOCG 2000), ACM Press, June 2000, pp. 43–49.
[36] G.T. Toussaint, R.S. Poulsen, Some new algorithms and software implementation methods for pattern recognition research, in: Proceedings of the IEEE International Computer Software Applications Conference, 1979, pp. 55–63.
[37] M. van Kreveld, J.S.B. Mitchell, P. Rousseeuw, M. Sharir, J. Snoeyink, B. Speckmann, Efficient algorithms for maximum regression depth, in: Proceedings of the 15th Annual Symposium on Computational Geometry (SOCG 1999), ACM, June 1999, pp. 31–40.