Algorithms for Bivariate Zonoid Depth

By
Harish Gopala

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario

September 2004

© Copyright
2004, Harish Gopala

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97451-0

Our file Notre référence

ISBN: 0-612-97451-0

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

PAGINATION ERROR.

ERREUR DE PAGINATION.

TEXT COMPLETE.

LE TEXTE EST COMPLET.

Abstract

Zonoid depth is a new notion of data depth proposed by Dyckerhoff et al. [16]. We give efficient algorithms for solving several computational zonoid depth problems for 2-dimensional (bivariate) data. These include algorithms for computing a zonoid depth map, computing a zonoid depth contour, and computing the zonoid depth of a query point.

Acknowledgements

I would like to acknowledge the help, support, guidance and blessings given to me by the following people.

Firstly, I bow, circumambulate, prostrate and place this piece of work at the lotus feet of that supreme arbiter, Lord Hari, who, through Sri Vaayu, has been guiding my thoughts and deeds all along. I thank them for listening to my ardent appeals and helping me out during times of crises. I beg them to continue guiding me through my endeavors, never letting me stray from the path of Dharma, and to give me opportunities to discharge my duties with pleasure.

Secondly, I give heartfelt thanks to my parents without whose wholehearted support, this thesis, nay, even graduate studies would have been impossible.

Thirdly, I thank my adviser Pat Morin for taking me in as his graduate student and for having so much faith and patience in me. By introducing me to Computational Geometry (CG), he has helped me find my calling, and for that I am forever grateful. He has put up with my procrastinations and resulting tardiness without so much as a remonstrance.

Fourthly, I thank my would-be advisers Anil Maheshwari and Michiel Smid for helping me find my way through difficult concepts in CG. They have helped me monetarily when the going was hard. They have acted as sounding boards for my often weird ideas.

Lastly, I thank Diwakar Krishnamurthy (Department of Electrical and Computer Engineering, University of Calgary), Paul McGee (Department of Geosciences, Princeton University) and Praveen Pai (Department of Electronics, Carleton University) for different things, too many to mention here.

Contents

1	Introduction	1
1.1	Definition of zonoid depth and regions	2
1.2	Other depth measures	2
1.3	Problems related to depth measures	3
1.4	Summary of results	4
1.5	Outline of thesis	5
2	Zonoids, k-sets and k-levels	6
2.1	Correspondence between zonoids and k -sets	6
2.2	Review of duality	8
2.3	Correspondence between zonoids and k -level	9
2.4	Some implications of the above observations	10
3	Algorithms for zonoids in 2 dimensions	11
3.1	Converting a k -level into a depth contour	11
3.2	Computing a depth contour	12
3.3	Computing a depth map	12
3.4	Testing if a zonoid contains a point	13
3.5	Computing the zonoid depth of a point	20
4	Conclusions and open problems	24
	Bibliography	25

List of Figures

2.1	1-sets on S and the 1-zonoid	7
2.2	2-sets on S and the 2-zonoid	8
2.3	k -zonoid in primal and dual	10
3.1	Vertical strip V showing the k -level and corresponding upper boundary of the dual of the k -zonoid	14
3.2	p_1^* is tangent to the upper boundary of the dual of the k -zonoid to the <i>right</i> of V	15
3.3	p_1^* is tangent to the upper boundary of the dual of the k -zonoid to the <i>left</i> of V	16
3.4	p_1^* is tangent to the upper boundary of the dual of the k -zonoid <i>inside</i> V	17
3.5	Figure showing how we build the trapezoid $ABB'A'$	18
3.6	Partitioning the $Z_k(S)$ zonoid into triangles	22

Chapter 1

Introduction

Data depth is a way of measuring how deep or central a given point x in $\mathbb{R}^d$ is with respect to a given data cloud $\{x_1, x_2, \dots, x_n\}$. This concept provides 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. [28] 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. However, many computational problems associated with such data depth functions are non-trivial to compute efficiently. Thus the study of these functions and related algorithms is essential for these functions to become more useful in statistics. Computational geometry [38] has been of great help in this regard, and there are many algorithms in the computational geometry literature giving efficient algorithms for data depth problems [1–5, 9–11, 13, 16, 19, 20, 23, 25–27, 32, 34, 36, 41, 42, 44, 45].

López-Pintado et al. [31] apply depth functions to the analysis of environmental data. Cheng et al. [12] apply techniques derived from data depth concepts on multivariate aviation safety data to develop meaningful threshold systems for both regulating and monitoring purposes. Jörnsten et al. [21, 22] apply the $L1$ depths to select clusters of genes in the human body in the study of different types of cancers.

Here, we focus on one particular depth measure, *zonoid depth*, introduced by Dyckerhoff et al. [16] and which is the topic of a book by Mosler [35].

1.1 Definition of zonoid depth and regions

Given a set of points $S = \{p_1, p_2, \dots, p_n\}$ in $\mathbb{R}^2$, the convex hull of S is the set of all points that can be expressed as convex combinations of S , i.e.,

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

However

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

is called the zonoid of depth k or the k -zonoid. Here, and throughout the thesis, $1 \leq k \leq n$ is an integer. Since a zonoid is defined by a finite set of linear constraints, it forms a convex polygon. Furthermore, for $k_1 > k_2$, the k_1 -zonoid is a subset of the k_2 -zonoid, hence $\{1, \dots, n\}$ -zonoids are a set of nested convex polygons. The n -zonoid $Z_n(S)$ is a single point, the mean of S . For other properties of zonoids, see Dyckerhoff et al. [16] and Mosler [35].

Now, the zonoid depth of a point p is defined as the maximum k for which p lies inside $Z_k(S)$. Dyckerhoff et al. [16] 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 $\lambda_1, \dots, \lambda_n$. 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 this algorithm is unclear and Dyckerhoff et al. [16] only give experimental results.

In this thesis, even though we concentrate on integer values of k , the results and definitions extend to any real value $1 \leq k \leq n$.

1.2 Other depth measures

In this section, we give terse descriptions of several of the most common depth measures. Detailed explanations can be found in the adjoining references.

Convex Hull Peeling Depth : The convex hull peeling depth at a sample point p with respect to the data set S is simply the level of the convex layer p belongs

to. The 0^{th} convex layer L_0 is the boundary of $CH(S)$. The i^{th} convex layer L_i is the boundary of $CH(S \setminus \cup_{j=0}^{i-1} L_j)$ [28].

Halfspace Depth : The halfspace depth of a query point p with respect to a set S of n points in $\mathbb{R}^d$ is defined as the minimum number of points of S in any closed halfspaces that contains p [28].

Hyperplane Depth : The hyperplane depth of a point p with respect to a set of n hyperplanes is defined to be the minimum number of hyperplanes that a ray emanating from p must cross [39].

$L1$ Median : The $L1$ median with respect to a set S of n points is the point which minimizes the sum of Euclidean distances to all points in a given data set [46].

Oja depth : Consider $d + 1$ points in $\mathbb{R}^d$. These points form a simplex, which has a d -dimensional volume. Now consider a data set in $\mathbb{R}^d$ for which we seek the median. Then Oja [37] proposed the following measure for a query point p in $\mathbb{R}^d$.

- for every subset of d points from the data set S , form a simplex with p .
- sum together the volumes of all such simplices.

This sum is called the Oja depth with respect to S .

Simplicial Depth : The simplicial depth of a point in $\mathbb{R}^d$ is the number of simplices whose vertices are elements of S that contain the point [29].

1.3 Problems related to depth measures

In this section, we present some standard computational problems related to depth measures.

Computing a depth contour : Given a set S of n points in $\mathbb{R}^2$ and an integer $1 \leq k \leq n$, construct a polygon containing exactly the points in the plane having depth at least k .

Computing a depth map : Given a set S of n points in $\mathbb{R}^2$, construct all depth contours of depth $1 \leq k \leq n$.

Testing if a contour contains a point : Given a set S of n points in $\mathbb{R}^2$, an integer $1 \leq k \leq n$ and a query point p , test whether p lies inside the depth contour of depth k .

Computing the depth of a point : Given a set S of n points in $\mathbb{R}^2$ and a query point p , compute the largest integer k for which p lies inside the depth contour of depth k .

Computing a point of maximum depth : Given a set of n points S in $\mathbb{R}^2$, compute the depth contour of maximum depth.

1.4 Summary of results

We describe a relationship between k -zonoids and k -sets. By exploring this relationship, we obtain the following algorithms:

1. an $O(n \log n + nk^{\frac{1}{3}})$ expected time algorithm to compute a zonoid depth contour,
2. an $O(n^2)$ algorithm to compute zonoid depth map,
3. a linear time algorithm to test whether a zonoid depth contour contains a point, and
4. a linear time algorithm to compute the zonoid depth of a point,

Algorithms 2, 3 and 4 are optimal. Algorithm 1 would require an improvement to the improved bounds on the maximum number of k -sets of a planar point set. This is a longstanding open problem due to Erdős et al. [18].

The results in this thesis have been presented in preliminary form at the 16th Canadian Conference on Computational Geometry [19].

Table 1.1 shows how our results compare with existing algorithms for other depth measures. All results in Table 1.1 are for 2-dimensional (bivariate) data.

Depth	Depth contour	Depth map	Depth of a point	Point/Region of maximum depth
Convex Hull Peeling	$O(n \log n)$ [10]	$O(n \log n)$ [10]	$O(n \log n)$ [10]	$O(n \log n)$ [10]
Halfspace	-	-	$O(n \log n)$ [40]	$O(n \log n)$ expected [9]
Hyperplane	-	-	-	$O(n \log n)$ [26]
$L1$ median	Only approximate algorithms are available			
Oja	-	-	$\Omega(n \log n)$ [3]	$O(n \log^3 n)$ [2]
Simplicial	-	-	-	$O(n^4)$ [2]
Zonoid	$O(n \log n + nk^{\frac{1}{3}})$ expected	$O(n^2)$	$O(n)$	$O(n)$

Table 1.1: Comparing running times of algorithms for different depth functions

For the $L1$ median, only approximation algorithms are available since the $L1$ median cannot be expressed as a rational value [6].

1.5 Outline of thesis

The remainder of this thesis is organized as follows.

In Chapter 2, we describe the relationship between zonoids, k -sets and k -levels.

In Chapter 3, we develop the algorithms stated in Section 1.4.

In Chapter 4, we restate our results and list some open problems pertaining to zonoid depth.

Chapter 2

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 thesis, point sets are always assumed to be in general positions.

Given a set S of n points in general position and an integer $0 \leq k \leq n - 2$, a set $S' \subseteq S$ is called a k -set if 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 Erdős et al. [18] 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 $O(nk^{\frac{1}{3}})$ is due to Dey [15].

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 Figure 2.1. By joining all such 1-set points, we get the zonoid of depth 1 or 1-zonoid or $Z_1(S)$, and $Z_1(S) = CH(S)$.

Now, on the same set S , construct all possible 2-sets. In each 2-set, take the mean (represented by the **X** on the dotted line segment joining 2 points in each 2-set as in Figure 2.2) of the pair of points. The following lemma shows that by joining all such means from all 2-sets of S , we obtain a 2-zonoid of S or $Z_2(S)$. In a similar fashion, zonoids up to depth n can be constructed.

 Figure 2.1: 1-sets on S and the 1-zonoid

Lemma 2.1.1. *Given a set S of n points in $\mathbb{R}^2$ and an integer $1 \leq k \leq n$, there is a bijection between the vertices of $Z_k(S)$ and the k -sets of S .*

Proof. We show that the relationship between k -zonoid vertices and k -sets is both one-to-one and onto.

One-to-one: Consider a k -set in a set S of n points. Allot $\lambda_i = \frac{1}{k}$ to each point in this k -set. This is in accordance to the definition of a k -zonoid in Section 1.1. Then $\sum_{i=1}^n \lambda_i p_i$ is the mean of the points in the k -set as well as an extreme point of $Z_k(S)$ in the direction perpendicular to the line that separates this k -set. There can be only one k -set that is extreme in any given direction, and the neighboring k -sets differ by at least one point. Therefore, different k -sets correspond to different k -zonoid vertices.

Onto: A k -zonoid vertex is extreme in some direction and is the mean of the points in the k -set separated by a line perpendicular to that direction. $\square$

Figure 2.2: 2-sets on S and the 2-zonoid

2.2 Review of duality

A point in the plane has 2 parameters: its x -coordinate and its y -coordinate. A non-vertical line in the plane also has 2 parameters: its slope and its y -intercept. Therefore, we can map a set of points to a set of lines and vice versa in a one-to-one manner. Such transforms are called *duality transforms* and the image of an object under a duality transform is called the *dual* of the object.

Let $p = (p_x, p_y)$ denote a point in the plane. The dual of p , denoted by p^* , is the line denoted by $p^* = \{(x, y) : y = p_x x - p_y\}$. The dual of the line $l = \{(x, y) : y = ax + b\}$ is the point $l^* = (a, -b)$.

The duality transform is not valid for vertical lines. This is not a problem because, in most cases, vertical lines can be handled separately, or the plane containing the lines can be rotated so that there are no vertical lines.

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 lines pass through a common point. For a detailed explanation on duality, see de Berg et al. [14].

2.3 Correspondence between zonoids and k -level

Let L be a set of n lines in the plane. The set L induces a subdivision of the plane that consists of vertices, edges and faces. Some of the edges are unbounded. This subdivision is usually referred to as the *arrangement* induced by L and is denoted by $A(L)$. Arrangements can be computed in $O(n^2)$ time by an incremental algorithm. The Zone theorem [17] implies that inserting the i^{th} line takes only $O(i)$ time.

We define the k -level as follows: Let L be a set of n lines in $\mathbb{R}^2$ that are dual to a set of n points P in general position in $\mathbb{R}^2$. Denote the line arrangement of L by $A(L)$. For $1 \leq k \leq n$, the k -level in $A(L)$ is the closure of all points on the given lines, which have exactly k lines on and above them.

The set of vertices of $A(L)$ with exactly k lines on and above them is denoted as S_k . Each vertex $v \in S_k$ is mapped to a dual line v^* that supports a k -set edge e_{pq} passing through two points p, q in P and has exactly $k - 2$ points above it.

We describe the k -zonoid in both the primal and the dual settings and show the relationship between them. The left part of Figure 2.3 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 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.

We constructed the k -zonoid in the primal by finding all possible k -sets, taking the mean of k points in each k -set and then joining these mean points of all k -sets. In the dual, we first construct the k -level (respectively the $(n - k)$ -level) and then, for each vertex which has k lines on and above it, we draw an upwards (respectively downwards) vertical ray through it, and compute the mean of the k lines that intersect

 Figure 2.3: k -zonoid in primal and dual

this vertical ray. Such mean points are then joined to get the boundaries of the shaded region in Figure 2.3. It may be observed here that for each vertex on the upper (respectively lower) boundary of the dual of the k -zonoid, there is a vertex directly below (respectively above) it on the k -level (respectively the $(n - k)$ -level).

2.4 Some implications of the above observations

Although the number of k -sets and the number of vertices of the k -level are different, they are within a constant fraction of each other [17]. Dey [15] proves an $O(nk^{\frac{1}{3}})$ 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 2.1.1, $O(nk^{\frac{1}{3}})$ is an upper bound on the number of vertices of a k -zonoid in $\mathbb{R}^2$.

Sharir et al. [43] show that the number of k -sets in a set of n points in $\mathbb{R}^3$ is $O(nk^{\frac{3}{2}})$. Because of this fact and Lemma 2.1.1, the maximum complexity of a zonoid in $\mathbb{R}^3$ is also $O(nk^{\frac{3}{2}})$.

Chapter 3

Algorithms for zonoids in 2 dimensions

In this chapter we present algorithms for zonoid depth. The first few algorithms follow easily from existing results on k -sets and arrangements. The later algorithms are somewhat more involved and require some sophisticated tools from computational geometry.

3.1 Converting a k -level into a depth contour

This section assumes that the k -level of the lines that are dual to the set S of n points in the primal has already been computed, since the construction of the k -level is described in Section 3.2.

Start walking on the k -level from left to right. At the first encountered vertex, take the mean of the k lines on and above this vertex. Now, for the remaining vertices on the k -level, do the following. If the k -level turns left at a vertex, then the mean computed for the previous vertex has changed because of the incidence of a new line on or above the current vertex. Therefore the mean has to be adjusted by discarding the line that is no longer one of the k lines on or above the current vertex and adding the new incident line. This adjustment takes constant time. If the k -level turns right at a vertex, then the mean does not change, so nothing needs to be done here.

Since this algorithm involves walking on the k -level and doing at most a constant time operation at each vertex of the k -level, the time it takes is $O(n)$ plus time linear in the number of vertices on the k -level. Dey [15] proves an $O(nk^{\frac{1}{3}})$ upper bound on the complexity of planar k -levels which is an upper bound for this algorithm as well.

3.2 Computing a depth contour

Chan [8] describes an algorithm for computing the k -level in a set of n lines that runs in $O(n \log n + nk^{\frac{1}{3}})$ expected time. The algorithm scans the k -level left to right using 2 kinetic priority queues. One priority queue contains the lines on levels $1, 2, \dots, k$ and the other contain lines on levels $k+1, k+1, \dots, n$. When the k and $k+1$ levels intersect, the top elements of the first queue is deleted and inserted into the second queue and vice-versa.

Using this algorithm, the k -level and the $(n-k)$ -level can be constructed in $O(n \log n + nk^{\frac{1}{3}})$ expected time. Once we have computed the k -level, we can compute the k -zonoid using the method outlined in Section 3.1.

Theorem 3.2.1. *Given a set S of n points in $\mathbb{R}^2$ and an integer $1 \leq k \leq n$, the k -zonoid (i.e. the zonoid depth contour of depth k) of S can be computed in $O(n \log n + nk^{\frac{1}{3}})$ expected time.*

3.3 Computing a depth map

Applying Theorem 3.2.1 n times for k from 1 to n gives us an algorithm to compute a depth map in $O(n^2 \log n + n^2 k^{\frac{1}{3}})$ expected time. But the relationship between k -zonoids, k -levels and $(n-k)$ -levels allows us to compute all $\{1, \dots, n\}$ -zonoids in $O(n^2)$ time by computing all arrangements of the dual lines [17]. Once we have the arrangement of lines, we can output the depth map in $O(n^2)$ time by applying the algorithm of Section 3.1 to one level at a time.

Theorem 3.3.1. *Given a set S of n points in $\mathbb{R}^2$, $\{1, \dots, n\}$ -zonoids (i.e. the depth map) can be computed in $O(n^2)$ time.*

Once we have computed the $\{1, \dots, n\}$ -zonoids, we can preprocess them for point locations using Kirkpatrick's planar point location algorithm [24] so that we can determine the zonoid depth of any point in $O(\log n)$ time.

Theorem 3.3.2. *Given a set S of n points in $\mathbb{R}^2$ and a query point p , after $O(n^2)$ preprocessing time we can construct an $O(n^2)$ space data structure such that the zonoid depth of p can be computed in $O(\log n)$ time.*

3.4 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 $1 \leq k \leq n$, report whether p lies inside or outside $Z_k(S)$.

Consider again Figure 2.3. 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, parallel to itself. When p_1 hits the k -zonoid boundary, p_1^* also hits the boundary of the dual of the k -zonoid. Since this boundary is convex, line p_1^* becomes tangent to it. This leads us to the following idea: in the primal, draw a vertical line through the point p_1 . It intersects the k -zonoid at 2 points (if the point p_1 is outside *and* to the right or to the left of the k -zonoid, then it is trivially outside and neglected). Finding these intersection points is equivalent to finding the points at which the 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 decided 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. [30]. In their algorithm, Lo et al. [30] 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 of the set of points and bisecting this set. But in our problem, we are searching for a vertex on the k -level that supports

the point on the upper boundary of the dual of the k -zonoid at which p_1^* is tangent. Nonetheless, both problems involve searching for a vertex on a k -level, therefore, we adopt their procedure.

Figure 3.1: Vertical strip V showing the k -level and corresponding upper boundary of the dual of the k -zonoid

Consider an open vertical strip V in the dual, as in Figure 3.1, showing the k -level and the corresponding convex upper boundary of the dual of the k -zonoid. To find out whether or not the line p_1^* is tangent to some boundary vertex inside V , we do the following: we count the first k lines from the top intersected by the left vertical line of V , and compute their mean. This actually involves finding the mean of the slopes and intercepts of these lines (which are the x - and y -coordinates of the points corresponding to these lines in the primal). This gives us the line containing the segment of the upper boundary of the dual of the k -zonoid intersected by the left vertical line of V . We do similarly for the right vertical line of V . Now we compare the slopes s_1 and s_2 of the left and right intersected segments respectively with slope s_p of line p_1^* . If $s_1, s_2 \leq s_p$ as in Figure 3.2, then p_1^* is a tangent to the upper boundary *to the right* of strip V . Similarly, if $s_1, s_2 \geq s_p$ as in Figure 3.3, then p_1^* is a tangent to the upper boundary *to the left* of strip V . But if $s_1 < s_p < s_2$ as in Figure 3.4, then p_1^* is a tangent to the boundary inside strip V . The running time of this check is $O(n)$, because we count k lines on the left and right vertical line and compute their

Figure 3.2: p_1^* is tangent to the upper boundary of the dual of the k -zonoid to the right of V

means. The slope comparison is, of course, done in constant time. Hence we have the following lemma.

Lemma 3.4.1. *Given a set S of n points in $\mathbb{R}^2$, a query point p and an integer $1 \leq k \leq n$, all in the primal, if V is an open vertical strip in the dual, we can find out whether or not this strip contains the boundary vertex of the dual of the k -zonoid at which the line p^* is tangent, in $O(n)$ time.*

Lo et al. [30] describe the following lemma:

Subdivision Lemma. *Let H be a set of n lines in the plane in general position, let $\alpha < 1$ be a prescribed positive constant and let V be an interval on the x -axis. In $O(n)$ time, V can be subdivided into subintervals $V_1, V_2, \dots, V_C$ ($C = C(\alpha)$ is a function that depends only on α), such that each V_i contains at most αN of the*

$$N = \binom{n}{2}$$

vertices of arrangement of H .

Figure 3.3: p_1^* is tangent to the upper boundary of the dual of the k -zonoid to the left of V

It is mentioned that $C(\alpha) \leq \frac{2}{\alpha}$.

This deterministic algorithm by Lo et al. [30] is quite complicated. An easier (randomized) method is to simply take a (suitably chosen) constant sized random sample of the vertices of the arrangement of lines. With constant probability, this will satisfy the conditions of the lemma.

In our problem, we apply the Subdivision Lemma in the dual. Let our strip V now be the entire plane so as to contain all intersections of n lines in the dual. We want to subdivide this strip into substrips so that each substrip contains at most $\frac{1}{20} \binom{n}{2}$ of the total number of intersections in V . The reason for this choice will become clear shortly. So we set $\alpha = \frac{1}{20}$ and get at most 40 subintervals $V_1, V_2, \dots, V_{40}$, because $C(\alpha) \leq \frac{2}{\alpha} = 40$.

We now have 40 subintervals or open vertical strips and we can find out, according to Lemma 3.4.1, in $O(n)$ time which V_i contains the upper boundary vertex at which line p_1^* is a tangent.

In V_i , we build a trapezoid $T = ABB'A'$, as shown in Figure 3.5. The points A

Figure 3.4: p_1^* is tangent to the upper boundary of the dual of the k -zonoid *inside* V

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 $k - \lfloor \frac{n}{6} \rfloor \leq 0$, then A and B are chosen below all lines, and similarly for the top side).

We now summarize our constructions in this section, because Lemmata 3.4.2 and 3.4.3 are directly based on this summary.

Let S be a set of n points and p be a query point in $\mathbb{R}^2$ in the primal, and let $1 \leq k \leq n$ be an integer. Subdivide the dual into subintervals $V_1, V_2, \dots, V_C$ ($C = C(\alpha)$ is a function that depends only on α), such that each V_i contains at most αN , $\alpha < 1$ of the $N = \binom{n}{2}$ vertices of the arrangement in the dual. Find the vertical strip V that contains the vertex of the boundary of the dual of the k -zonoid at which p^* becomes a tangent. Construct a trapezoid T in V bounded from the sides by the two vertical walls of V , from the top by the line segment joining two points each just below the $k + \lfloor \frac{n}{6} \rfloor$ level and one on each vertical boundary of V , and from the bottom by the line segment joining two points each just below the $k - \lfloor \frac{n}{6} \rfloor$ level and one on each vertical boundaries of V . Then we have the following Lemma.

Lemma 3.4.2. *The top and bottom sides of T are intersected by at most $\frac{n}{3}$ lines each.*

Proof. Let u be the number of lines intersecting the side AB of T upwards, i.e., these

Figure 3.5: Figure showing how we build the trapezoid $ABB'A'$

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 intersections. But we created each strip V_i such that it has at most $\frac{1}{20} \binom{n}{2}$ intersections. Therefore,

$$\begin{aligned}
 ud &\leq \frac{1}{20} \binom{n}{2} \\
 \Rightarrow ud &\leq \frac{1}{20} \frac{n(n-1)}{2} \\
 \Rightarrow ud &\leq \frac{n^2}{40} \\
 \text{But } ud &= u^2 \quad (\because u = d) \\
 \text{So } u^2 &\leq \frac{n^2}{40}
 \end{aligned}$$

$$\begin{aligned}
 \Rightarrow u &\leq \frac{n}{\sqrt{40}} \\
 \Rightarrow u &\leq \frac{n}{6.325} \quad (\because \sqrt{40} \approx 6.325) \\
 \Rightarrow u &\leq \frac{n}{6} \\
 \Rightarrow d &\leq \frac{n}{6} \\
 \Rightarrow u + d &\leq \frac{n}{3}
 \end{aligned}$$

But $u + d$ is the number of lines intersecting side AB of trapezoid T . A similar argument holds for side $A'B'$ as well. $\square$

The 2 vertical sides of the trapezoid T are also intersected by at most $\frac{n}{3}$ lines each, because of our choice of the levels of the points A, B, B' and A' . So each side of T is intersected by $\leq \frac{n}{3}$ lines (which is exactly why we chose $\alpha = \frac{1}{20}$ in the Subdivision Lemma), hence we have at most $\frac{4n}{3}$ intersections altogether. Each line that intersects the trapezoid contributes 2 to this sum, so we have at most $\frac{2n}{3}$ lines that intersect the trapezoid, which means $\frac{n}{3}$ lines pass outside the trapezoid.

Also based on the summary of constructions given just before Lemma 3.4.2, we have the following Lemma.

Lemma 3.4.3. *Within V , the k -level is completely contained within T .*

Proof. Suppose that the k -level breaks below the side AB of T . Then some point C on this side has level greater than k . But since both A and B have level $k - \lfloor \frac{n}{6} \rfloor$, both the segments AC and BC have to be intersected by more than $\frac{n}{6}$ lines. But this contradicts Lemma 3.4.2, which says that AB is intersected by $\leq \frac{n}{3}$ lines. $\square$

Lemma 3.4.3 holds even if the strip V is bounded only on one side. Either the point A or B is at infinity. Based on Lemmata 3.4.2 and 3.4.3, 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.

It may be observed that until recently in our discussion, we were trying to find the vertex on the boundary of the dual of the k -zonoid at which line p_1^* becomes a tangent. But above we show that it is the k -level that lies completely inside the trapezoid T and the vertex point is being searched for on the k -level, not on the boundary of the dual of the k -zonoid. This is because,

1. we do not explicitly construct the k -zonoid in the dual,
2. each vertex on the boundary of the dual of the k -zonoid is calculated by taking the mean of the k lines above a k -level vertex,
3. even though we disregard lines above and passing outside the trapezoid T , their equations can be associated with each k -level vertex and the corresponding k -zonoid dual vertex be computed, as need arises.

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. In this manner, each time we remove a constant fraction of the existing number of lines. Hence the algorithm runs in $O(n)$ time. So we have the following theorem.

Theorem 3.4.4. *Given a set S of n points in $\mathbb{R}^2$, a query point p and an integer $1 \leq k \leq n$, we can find out 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.*

3.5 Computing the zonoid depth of a point

To compute the depth of a point we make use of a general technique due to Chan [7] which requires

1. a decision algorithm to decide whether the solution is smaller than some value in $\mathbb{R}$ in time proportional to the size of the problem, and

2. that, from the problem, r subproblems can be constructed, each of size at most a fraction of the size of the main problem, within time proportional to the size of the main problem and the overall solution is the minimum or maximum of the solutions of these r subproblems.

If these conditions are met, then Chan's technique [7] allows us to solve the main problem in time proportional to its size. Chan's technique [7] is based on the observation that, if we consider the subproblems in random order, then the expected number of updates to the maximum (or minimum) is at most $\ln r + 1$. Determining when to update the maximum can be done using the (inexpensive) decision algorithm while actually updating the maximum is done with an (expensive) recursive call.

The decision algorithm comes from Section 3.4. We now describe the decomposition of our problem into subproblems.

Given a set of points $S = \{p_1, p_2, \dots, p_n\}$ in general position and a set of positive weights $w = \{w_1, w_2, \dots, w_n\}$ where $W = \sum_{i=1}^n w_i$, a *weighted zonoid* of depth k is defined as

$$Z_k(S, w) = \left\{ \frac{1}{W} \sum_{i=1}^n \lambda_i w_i p_i \mid 0 \leq \lambda_i \leq \frac{1}{k}, \sum_{i=1}^n \lambda_i = 1 \right\}$$

Note that this is a generalization of zonoid depth where a weight or multiplicity w_i is attached to each point p_i . Theorem 3.4.4 extends to the weighted case without much difficulty. The main modification is that the definition of a level is changed to take weights into account.

We partition our problem into 4 subproblems S_1, S_2, S_3 and S_4 as follows: we first partition the set S of n points into 4 quadrants Q_1, Q_2, Q_3 and Q_4 , each containing roughly $\frac{n}{4}$ points, using Megiddo's algorithm [33]. Subproblem S_1 contains 3 consecutive quadrants, say Q_1, Q_2, Q_3 , and a single point whose weight is the weighted average of all the points in Q_4 . So S_1 has $\frac{3n}{4} + 1$ points. We define the sets S_2, S_3 and S_4 in a similar manner. We define $\text{depth}(p, S_i)$ as the zonoid depth of point p in problem S_i . Note that this merging produces a strictly smaller zonoid, i.e. $Z_k(S_i, w) \subseteq Z_k(S)$.

Lemma 3.5.1.

- a. $\text{depth}(p, S_i) \leq \text{depth}(p, S)$ for each $1 \leq i \leq 4$,

$$b. \text{depth}(p, S) = \max\{\text{depth}(p, S_i) \mid 1 \leq i \leq 4\}$$

 Figure 3.6: Partitioning the $Z_k(S)$ zonoid into triangles

Proof. Part *a* follows from the observation that $Z_k(S_i) \subseteq Z_k(S)$. To see why Part *b* is true, suppose $\text{depth}(p, S) = k$ 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 Figure 3.6. 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 quadrants $Q_1, \dots, Q_4$. Wlog suppose they miss Q_4 . Then it is not hard to see that $Z_k(S_1)$ has a and b as vertices. Furthermore, $Z_k(S_1)$ contains $Z_n(S)$ and is convex, so it also contains p . Therefore $\text{depth}(p, S_1) \geq k = \text{depth}(p, S)$ as required. $\square$

This satisfies the requirements of Chan's optimization algorithm [7] because:

1. we have a decision algorithm that can decide whether the solution is smaller than an integer k in time proportional to the size of the problem, i.e. linear, as shown by Theorem 3.4.4,

2. we can construct the subproblems, whose sizes are a fraction of the size of the problem, in linear time, and
3. the overall solution is the maximum of the solutions of the subproblems, as shown by Lemma 3.5.1.

Thus, we have the following theorem.

Theorem 3.5.2. *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 $Z_k(S)$, in $O(n)$ time.*

Chapter 4

Conclusions and open problems

In Chapter 1, we gave the definitions of different zonoid depth related terms such as k -zonoid $Z_k(S)$ of a set S of n points and zonoid depth of a point p . Then we listed some problems related to depth measures and our results for those problems pertaining to zonoids.

In Chapter 2, we described the relationship between zonoids, k -sets and k -levels.

In Chapter 3, we gave:

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

Algorithms 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 thesis only deals with zonoid depth problems in 2 dimensions and, besides the paper by Dyckerhoff et al. [16], no work has been done regarding zonoid depth problems in dimensions 3 and higher. Therefore, handling zonoids in higher dimensions remains an open problem.

Bibliography

- [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, and G. Toussaint. Algorithms for bivariate medians and a Fermat-Torricelli problem for lines. *Computational Geometry: Theory and Applications*, 26(1):69–79, August 2003.
- [3] G. Aloupis and E. McLeish. A lower bound for computing Oja depth. Manuscript, School of Computer Science, McGill University, Montreal, Canada, 2004.
- [4] G. Aloupis, M. Soss, and 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, and S. Teng. Regression depth and center points. *Discrete and Computational Geometry*, 23(3):305–323, March 2000.
- [6] C. Bajaj. Proving geometric algorithm non-solvability: An application of factoring polynomials. *Journal of Symbolic Computation*, 2(1):99–102, March 1986.
- [7] T. Chan. Geometric applications of a randomized optimization technique. *Discrete and Computational Geometry*, 22(4):547–567, December 1999.
- [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)*, pages 430–436. SIAM, January 2004.
- [10] B. Chazelle. On the convex layers of a planar set. *IEEE Transactions on Information Theory*, IT-31(4):509–517, July 1985.
- [11] A. Cheng and M. Ouyang. On algorithms for simplicial depth. Technical Report dcs-tr-368, Department of Computer Science, Rutgers University, 1998.
- [12] A. Y. Cheng, R. Y. Liu, and J. T. Luxhøj. Monitoring multivariate aviation safety data by data depth: control charts and threshold systems. *IIE Transactions on Operations Engineering*, 32(9):849–859, September 2000. Special Issue on Probabilistic Models for Design, Planning, and Control of Production and Service Systems.
- [13] R. Cole, M. Sharir, and C. K. Yap. On k -hulls and Related Problems. *SIAM Journal on Computing*, 16(1):61–77, February 1987.
- [14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwartzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.
- [15] 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)*, pages 156–161. IEEE Computer Society, October 1997.
- [16] R. Dyckerhoff, G. Koshevoy, and K. Mosler. Zonoid data depth: Theory and computation. In A. Prat, editor, *COMPSTAT 1996 - Proceedings in Computational Statistics*, pages 235–240. Physica-Verlag, Heidelberg, August 1996.
- [17] H. Edelsbrunner. *Algorithms in combinatorial geometry*. Springer-Verlag New York, Inc., 1987.
- [18] P. Erdős, L. Lovász, A. Simmons, and E. G. Straus. Dissection graphs of planar point sets. In *A Survey of Combinatorial Theory*, pages 139–149. North-Holland, 1973.

- [19] H. Gopala and P. Morin. Algorithms for Bivariate Zonoid Depth. In *Proceedings of the 16th Canadian Conference on Computational Geometry (CCCG 2004)*, pages 132–135, August 2004.
- [20] S. Jadhav and A. Mukhopadhyay. Computing a Centerpoint of a Finite Planar Set of Points in Linear Time. *Discrete and Computational Geometry*, 12:291–312, 1994.
- [21] R. Jörnsten. Clustering and classification based on the L_1 data depth. *Journal of Multivariate Analysis*, 90(1):67–89, July 2004.
- [22] R. Jörnsten, Y. Vardi, and C.-H. Zhang. A Robust Clustering Method and Visualization Tool Based on Data Depth. In Y. Dodge, editor, *Statistical Data Analysis Based on the L_1 -Norm and Related Methods*, volume 12 of *Statistics for Industry and Technology*. Birkhäuser, 2002.
- [23] S. Khuller and J. S. B. Mitchell. On a Triangle Counting Problem. *Information Processing Letters*, 33(6):319–321, February 1990.
- [24] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, February 1983.
- [25] S. Langerman. *Algorithms and Data Structures in Computational Geometry*. PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, May 2001.
- [26] S. Langerman and W. Steiger. The Complexity of Hyperplane Depth in the Plane. *Discrete and Computational Geometry*, 30(2):299–309, August 2003.
- [27] S. Langerman and W. Steiger. Optimization in arrangements. In H. Alt and M. Habib, editors, *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2003)*, Lecture Notes in Computer Science, pages 50–61. Springer-Verlag, February 2003.

- [28] R. Liu, J. M. Parelius, and K. Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–858, June 1999.
- [29] R. Y. Liu. On a Notion of Data Depth Based on Random Simplices. *The Annals of Statistics*, 18(1):405–414, March 1990.
- [30] C.-Y. Lo, J. Matousek, and W. Steiger. Algorithms for ham-sandwich cuts. *Discrete and Computational Geometry*, 11:433–452, 1994.
- [31] S. López-Pintado and J. Romo. A functional depth analysis for environmental data. In *Proceedings of the ISI International Conference on Environmental Statistics and Health*, July 2003.
- [32] J. Matoušek. Computing the Center of Planar Point Sets. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: papers from the DIMACS special year*, volume 6, pages 221–230. American Mathematical Society, Providence, 1991.
- [33] N. Megiddo. Partitioning with two lines in the plane. *Journal of Algorithms*, 6(3):430–433, September 1985.
- [34] K. Miller, S. Ramaswami, P. Rousseeuw, T. Sellarès, D. Souvaine, I. Streinu, and A. Struyf. Fast implementation of depth contours using topological sweep. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 690–699. SIAM, January 2001.
- [35] K. Mosler. *Multivariate Dispersion, Central Regions and Depth. The Lift Zonoid Approach*, volume 165 of *Lecture Notes in Statistics*. Springer-Verlag New York, Inc, 2002.
- [36] A. Niinimaa, H. Oja, and J. Nyblom. Statistical Algorithms: Algorithm AS 277: The Oja Bivariate Median. *Journal of Applied Statistics*, 41(3):611–617, 1992.
- [37] H. Oja. Descriptive statistics for multivariate distributions. *Statistics and Probability Letters*, 1:327–332, 1983.

- [38] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- [39] P. J. Rousseeuw and M. Hubert. Depth in an Arrangement of Hyperplanes. *Discrete and Computational Geometry*, 22(2):167–176, 1999.
- [40] P. J. Rousseeuw and I. Ruts. Bivariate location depth. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 45:516–526, 1996.
- [41] P. J. Rousseeuw and I. Ruts. Constructing the bivariate Tukey median. *Statistica Sinica*, 8(3):827–839, July 1998.
- [42] M. Shamos. Geometry and statistics: problems at the interface. In *Proceedings of the Symposium on Algorithms and Complexity, Carnegie-Mellon University, Pittsburgh, PA.*, pages 251–280. Academic Press, New York, 1976.
- [43] M. Sharir, S. Smorodinsky, and G. Tardos. An improved bound for k-sets in three dimensions. In *Proceedings of the 16th Annual Symposium on Computational Geometry (SOCG 2000)*, pages 43–49. ACM Press, June 2000.
- [44] G. T. Toussaint and R. S. Poulsen. Some new algorithms and software implementation methods for pattern recognition research. In *Proceedings of the IEEE International Computer Software Applications Conference*, pages 55–63, 1979.
- [45] M. van Kreveld, J. S. B. Mitchell, P. Rousseeuw, M. Sharir, J. Snoeyink, and B. Speckmann. Efficient algorithms for maximum regression depth. In *Proceedings of the 15th Annual Symposium on Computational Geometry (SOCG 1999)*, pages 31–40. ACM, June 1999.
- [46] A. Weber. *Über den standort der industrien*, Tübingen. English translation by C. Friedrich (1929): Alfred Weber’s theory of location of industries, 1999. University of Chicago Press.