DIMACS Technical Report 99-57
December 1999

An Optimal Algorithm for Hyperplane Depth in the Plane

by

Stefan Langerman[1] William Steiger[2]

[1]Computer Science, Rutgers University
[2]Computer Science, Rutgers University

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Telcordia Technologies (formerly Bellcore) and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

Given a set of n hyperplanes $h_1, \ldots, h_n \in R^d$ the *hyperplane depth* of a point $P \in R^d$ is the minimum number of hyperplanes that a ray from P can meet. The *hyperplane depth* of the arrangement is the maximal depth of points P not in any h_i. We give an optimal $O(n \log n)$ deterministic algorithm to compute the hyperplane depth of an arrangement in dimension $d = 2$.

1 Introduction and Summary

Given a set $S = \{h_1, \ldots, h_n\}$ of n real numbers, the depth of $x \in R$ is defined to be

$$(1) \quad d(x) = \min(|\{h_i \leq x\}|, |\{h_i \geq x\}|),$$

and may be thought of as the "size" of the smallest halfline containing x. A median is a point of maximal depth ($\lceil (n+1)/2 \rceil$) and may be found in $O(n)$ time.

There are many situations - for example in Statistics - where multivariate generalizations of ranks and order statistics are useful, and several proposals for depth in R^d have been made [2], [7]. In one of the most familiar ones [14], we are given a set $S = \{P_1, \ldots, P_n\}$ of n points in R^d. The depth, or Tukey depth of $x \in R^d$ is defined as the minimal number of points of S that must be in a closed halfspace containing x. A median is a point of maximal depth. When $d = 1$ this notion agrees with the depth in (1). It is a well known consequence of Helly's Theorem (e.g.,[6]) that a Tukey median always has depth at least $\lceil n/(d+1) \rceil$. Matoušek [10] has described an algorithm for $d = 2$ that finds a median in time $O(n(\log n)^5)$. Cole, Sharir and Yap also gave an $O(n(\log n)^5)$ algorithm [5], which was further improved by Cole [3] to $O(n(\log n)^3)$.

Recently, Rousseeuw and Hubert [11] proposed an interesting, new notion of depth for R^d. Given a set $S = \{h_1, \ldots, h_n\}$ of n hyperplanes in R^d, they defined the *hyperplane depth* of a point $x \in R^d$ to be

$$(2) \qquad \delta(x) = \min_{u:\|u\|=1} (r(x,u)),$$

where $r(x,u)$ is the number of $h_i \in S$ that meet the ray $\{x + tu, t \geq 0\}$ through x in direction u. A median is a point of maximal depth. When $d = 1$, (2) agrees with the usual definition of depth in (1).

Hyperplane depth was motivated by problems in robust regression. Using a familiar point/hyperplane duality transformation, the h_i dualize to n data points in R^d and x, to a hyperplane meant to fit the data. The "regression depth" of x in is the minimum number of data points x meets in a rotation-to-vertical; this corresponds exactly to the hyperplane depth of x in the original problem.

Hyperplane depth has attracted much recent interest. On the combinatorial side it was just shown [1] that like the Tukey median, the hyperplane median must have depth at least $\lceil n/(d+1) \rceil$; the lower bound is sharp, attained when the h_i are duals of n points on the moment curve in R^d. Other interesting combinatorial questions remain open.

On the computational side, a main question is to determine the complexity of selection - computing points of given depth, or of maximal depth. We use the unit cost RAM model. The hyperplanes in S partition R^d into a complex of $O(n^d)$ convex cells, called the *arrangement* of S, and written $A(S)$. It is clear that every point in a cell has the same depth. The depth of the arrangement, $\delta(A(S))$ is the depth of the deepest cell. The task therefore is to compute $\delta(A(S))$, and a witness point of that depth, or to find a cell of given depth $k \leq \delta(A(S))$.

Rousseeuw and Hubert point out that for any x, $\delta(x)$ can be computed in $O(n^{d-1} \log n)$, and since there are $O(n^d)$ vertices in $A(S)$, selection has cost $O(n^{2d-1} \log n)$. For $d = 2$ they mention an $O(n^3)$ algorithm [12]. In Amenta et. al. [1] it was observed that $A(S)$ can be constructed in $O(n^d)$ and then, using breadth-first-search on the graph of adjacent cells, the depth of every cell is obtained in the same $O(n^d)$ time. This is the analogue of using sorting to find the ordinary median.

For the case $d = 2$, vanKreveld et.al. [9] recently described a beautiful $O(n(\log n)^2)$ algorithm for hyperplane depth in R^2. This shows that it is not necessary to find the depth of every cell in order to find the deepest one and is analogous to the fact that sorting is not optimal for the usual median. The algorithm is controlled by a binary search based on being able to select the k^{th} vertex - ordered by x-coordinate - in a set of consecutive, unsearched candidate vertices. This may be done in $O(n \log n)$ (see e.g., [4]). The vertical line through the selected vertex is then analyzed for *sidedness*. The sidedness test for a line ℓ determines one of the two half planes delimited by ℓ that intersects a maximum depth cell, and has complexity $O(n \log n)$. Since there are $O(\log \binom{n}{2})$ binary search steps, the algorithm has complexity $O(n \log^2 n)$.

In the present paper we prove the following.

Theorem 1 *Given a set S of n lines in the plane, the hyperplane depth of the arrangement, $\delta(A(S))$, and a witness point μ with that depth can be obtained in $\Theta(n \log n)$ time and $O(n)$ space.*

The proof is via an algorithm, shown to have the asserted complexity. The lower bound is via a connected components argument showing that $\Omega(n \log n)$ steps are needed by an algebraic decision tree that can decide if the depth of a given point is greater than k, a given integer. An easy consequence is that in the same time bounds, given $k \leq \delta(A(S))$, a point of depth k can be found.

It is interesting that our algorithm combines parametric search and *prune-and-search* to improve the result in [9], much like the way that the linear-time algorithm for the usual median improves on sorting. We are searching for a point μ in a cell C of maximal depth.

In fact, we will determine for each line in S its above/below relationship with μ. Each step of the algorithm solves the parametric problem of finding that above/below relationship for a constant fraction $a < 1$ of the m (initially n) *remaining* lines. These lines are then pruned out and the search continues with the remaining lines. We are able to do the pruning step in $O(m \log n + n)$. Since there are at most $n(1-a)^j$ unpruned candidates after the j^{th} step, and at most $\log n$ steps in all, the complexity is at most

$$cn \log n + \sum_{j=0}^{\infty} b(1-a)^j n \log n = O(n \log n).$$

2 Tools

In this section, we will describe some tools that will be used in the main algorithm. In particular, we will talk about the *sidedness* test for a line, and *insideness* test for a convex closed curve. In order to simplify the discussion, we define $\tilde{r}(x, u)$ to be the number of lines of S crossed by the ray $\{x + tu, t > 0\}$ (the difference with r being that we are not counting lines passing through x). We also define $\tilde{\delta}(x) = \min_{u:\|u\|=1}(\tilde{r}(x, u))$. Note that if x is inside a cell, $\delta(x) = \tilde{\delta}(x)$, and so, since we are looking for a cell of maximum depth, we can use $\tilde{\delta}$ instead of δ without changing the result.

Given a point x, we say that u is a *witness direction* if $\tilde{r}(x, u) = \tilde{\delta}(x)$. We will need the following:

Lemma 1 (Wedge Lemma [9]) *Let x be a point and u and v be directions of rays from x. Then no cell of of $A(S)$ intersecting the wedge (cone) formed by u and v at x has depth greater than $\max(\tilde{r}(x, u), \tilde{r}(x, v))$.*

Given a point x, we will call the *wedge at* x the set of all directions that can be formed by positive linear combinations of witness directions at x. By the Wedge Lemma, no cell that intersects the wedge at x can have depth greater than $\tilde{\delta}(x)$. Note also that if the wedge at x contains all directions, then x is of maximum depth. Otherwise, the wedge at x forms an angle less than π. Here are a few more facts worth noticing:

- If x is on a segment separating two adjacent cells and x^+ and x^- are points in the adjacent cells, then $\tilde{\delta}(x) = \min(\tilde{\delta}(x^+), \tilde{\delta}(x^-))$.
- On any path in the plane, points $x \neq y$ that have no line of S separating them are in the same cell, and have the same wedges. On any path in the plane (not touching any vertex of the arrangement), any 2 consecutive wedges intersect.

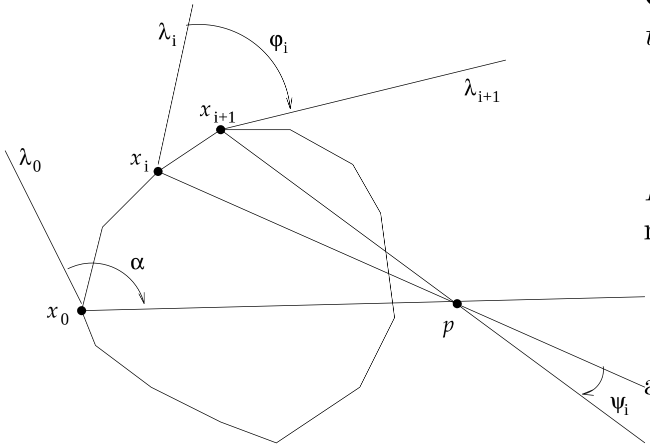
Using the Wedge Lemma, VanKreveld et.al. [9] described a sidedness test for a line ℓ. In time $O(n \log n)$ it computes $k = \max(\delta(x), x \in \ell)$ and gives a side (Left or Right) of ℓ which does not contain a point of depth $> k$.

We now describe a new way to test sidedness based on the following *insideness test.* Suppose we are given a convex polygon $\mathcal{P}$ whose boundary does not contain a vertex of $A(S)$, and such that no vertex of $\mathcal{P}$ is on a line of the arrangement. Let k denote the depth of the deepest point on the boundary of $\mathcal{P}$. We want to determine which of the inside or the outside of the convex polygon cannot contain a cell of depth greater than k. Note that if $k = \delta(A(S))$ is the depth of the arrangement, any answer would be correct.

Lemma 2 *Given a convex polygon $\mathcal{P}$ with s sides and a subset $U = \{\ell_1, \ldots, \ell_m\} \subseteq S$ of the lines of S that meet the interior of $\mathcal{P}$, then the insideness test for $\mathcal{P}$ can be performed in $O(m \log(n) + s)$.*

We will continuously move a point x around the boundary of $\mathcal{P}$, starting at the leftmost vertex, and keeping track of the wedge at x. Note that the wedge only changes when x crosses a line of U. In [9], a data structure is presented to maintain the wedge at x as x moves along a path. Its cost is $O(\log(n))$ per line crossing. At the same cost it gives $\max(\delta(x))$ for the points traversed, so k, the greatest depth of points on $\mathcal{P}$, is computed in the traversal of $\mathcal{P}$. The number of wedges is $4m$ since each line crosses the boundary of the polygon twice, and there are two wedges per line crossing, one when x moves onto a new line, and another when it passes that line. Let $x_0, \ldots, x_{4m-1}, x_{4m} = x_0$ be $4m$ points on the path that mark the wedge change events.

Observe the movement of the leftmost ray λ_i of the wedge at x_i along the path, and let $-\pi \leq \varphi_i \leq \pi$, $i = 0, \ldots, 4m-1$ be the angle between λ_i and λ_{i+1}. Note that since any two consecutive wedges intersect, we can move the ray to its next position while always staying inside a wedge, and any point swept by the ray cannot have depth greater than k.



Claim 2.1 *If the outside of the polygon contains a point of depth greater than k, then*

$$\sum_{i=0}^{4m-1} \varphi_i = 0$$

Proof. If p is a point outside of the polygon of depth greater than k, then that point is not swept by the ray, by the wedge lemma. Let $0 < \alpha < 2\pi$ be the angle $\lambda_0 x_0 p$, and let $-\pi \leq \psi_i \leq \pi$ be the angle between the two directions $x_i p$ and $x_{i+1} p$. By induction, for any r, we have

$$\alpha - 2\pi + \sum_{i=0}^{r} \psi_i < \sum_{i=0}^{r} \varphi_i < \alpha + \sum_{i=0}^{r} \psi_i$$

because the last movement of the ray cannot sweep p. Since p is outside the polygon, and $x_{4m} = x_0$, we have

$$\sum_{i=0}^{4m-1} \psi_i = 0$$

and so

$$-2\pi \leq \alpha - 2\pi < \sum_{i=0}^{4m-1} \varphi_i < \alpha \leq 2\pi$$

But $\sum_{i=0}^{4m-1} \varphi_i$ must be a multiple of 2π since $\lambda_{4m} = \lambda_0$, so it must be 0. ∎

Claim 2.2 *If the inside of the polygon contains a point of depth greater than k, then*

$$\sum_{i=0}^{4m-1} \varphi_i \neq 0$$

Proof. Same as for the previous claim, but now, since p is inside the polygon, we have

$$\sum_{i=0}^{4m-1} \psi_i = 2\pi$$

and so,

$$0 \leq \alpha - 2\pi + 2\pi < \sum_{i=0}^{4m-1} \varphi_i < \alpha + 2\pi$$

Thus the sum cannot be 0. ∎

Proof of Lemma 2.2. We can precompute the ordering of all $2m$ line crossings, walk around the polygon and compute k and $\Sigma\varphi_i$ in $O(m\log(n) + s)$ time. If the sum is 0, then the inside does not contain a point of depth greater than k. Otherwise, the outside does not contain a point of depth greater than k, and this proves the lemma. ∎

Corollary 1 *Suppose we have a convex polygon $\mathcal{P}$ with s sides, a subset $U = \{\ell_1, \ldots, \ell_m\} \subseteq S$ of the lines of S that meet the interior of $\mathcal{P}$, and a line ℓ. If $\mathcal{P}$ contains a cell C of maximal depth, then the sidedness test for ℓ can be performed in $O(m\log(n)+s)$.*

Proof. If ℓ does not meet $\mathcal{P}$, the sidedness test is trivial. Otherwise let $\mathcal{P}^+$ denote the part of $\mathcal{P}$ “above” ℓ, together with the segment on ℓ that meets $\mathcal{P}$. The insideness test for $\mathcal{P}^+$ resolves the sidedness of ℓ. ∎

Note that if the polygon is chosen to be big enough to contain all the vertices of the arrangement, this test performs the usual sidedness test from [9].

3 Pruning

At every step of the algorithm, the lines are divided into two sets: the set P contains all the pruned lines, and U contains the unpruned lines. Write $m = |U|$ and $A(U)$ for the arrangement of unpruned lines. For every line in P, we know if it is above or below some cell of maximal depth, and we maintain the intersection of these halfspaces as a convex polygon $\mathcal{P}$ which contains a cell of maximal depth. At the beginning of the algorithm, P is empty, and $\mathcal{P}$ is some polygon large enough to contain all the vertices in the arrangement (such a polygon can be constructed in $O(n \log n)$ time).

Consider some point μ in $\mathcal{P}$ of maximal depth, and let v be the vertical line passing through μ. We want to solve the parametric problem of determining for a constant fraction of the lines in U, whether those lines cross v above or below μ.

First, we compute the *centerpoint line* ℓ for U. This line is the dual of a (usual) centerpoint for the points dual to the lines in U. It has the property that it is below the $m/3$ level of $A(U)$ and above the $2m/3$ level. Jadhav and Mukhopadhyay [8] give a linear time algorithm to compute ℓ.

Next, perform the sidedness test for that line ℓ, restricted to the polygon $\mathcal{P}$, in $O(m \log(n) + n)$ time. Assume without loss of generality that μ is below ℓ. In $O(m)$ time, compute the intersection of all lines in U with ℓ, and select 9 of those intersection points, say $p_1, \ldots, p_9$, each adjacent pair separated by $\leq m/9$ of the intersections on ℓ. We can do this in $O(m)$ using the fast selection algorithm.

Perform the sidedness test (restricted to $\mathcal{P}$) for the vertical lines at each of those 9 points, in $O(m \log(n) + n)$ time. This will either find a point of maximal depth, or determine a strip between adjacent lines that contains the maximal depth point μ.

Let $x = a$ and $x = b$ denote the left and right verticals of that strip, $a < b$. By construction ℓ has at most $m/9$ intersections in $[a, b]$ with lines in U. In addition there are at least $m/3$ lines in U that meet $x = a$ or $x = b$, or both, above the level $m/3$. If these lines meet ℓ outside $[a, b]$ they cross v above μ and can be pruned. It follows that at least $m/3 - m/9 = 2m/9$ lines of U can be pruned, since they have been determined to be above μ.

In order to prune a line, we remove it from U and add it to P. We then update the polygon $\mathcal{P}$ in $O(\log(n))$ per line using an incremental convex hull algorithm. The overall pruning cost is therefore $O(m \log(n) + n)$. When the number of unpruned lines falls below some predefined constant, we can perform the sidedness test to all remaining lines in $O(n \log n)$ time, and that completes the description of the algorithm.

A simpler algorithm would replace the centerpoint-line construction that used Jadhav and Mukhopadhyay's algorithm by a random choice of a line from U. With positive constant probability, the intersection of that line with v, the vertical line through μ, will have $n/3$ intersections above and below it. The resulting randomized algorithm would have the same $O(n \log n)$ running time for its expected cost.

4 The Lower Bound

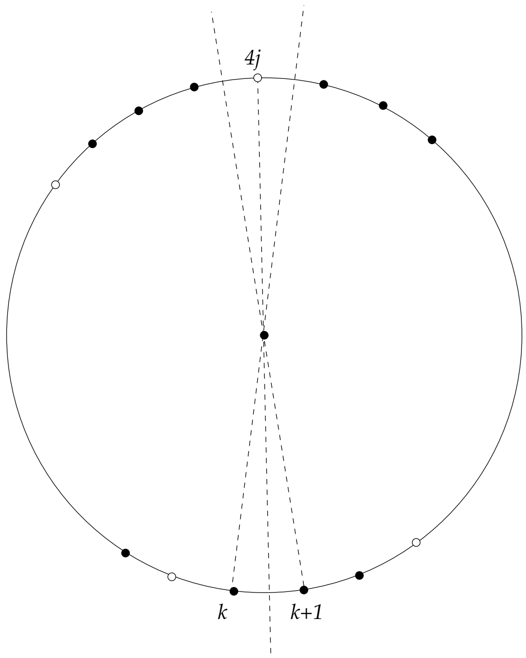
Let C denote the circumference of the unit circle in R^2 and let T denote an algebraic decision tree that can decide whether $\delta(A(S)) \geq \lfloor m/2 \rfloor$ for a set S of m lines in R^2. We restrict attention to lines that are tangent to

C and we will even hold most of the lines fixed. Such inputs can be encoded by the arguments $(\phi_1, \ldots, \phi_m) \in R^m$ of the tangency points. We will show that the subset $Y \subseteq R^m$ of restricted inputs where T returns YES has $\Omega(m \log m)$ path connected components.

Consider the point $\theta = (\theta_1, \ldots, \theta_m)$ where

(3) $$\theta_i = \frac{i}{4n+3} 2\pi,\ i = 1, \ldots, 4n+3.$$

This describes $m = 4n + 3$ equally spaced points on C, no two diametric. In fact the diameter containing θ_i is a halving line - both its open semicircles contain $2n+1$ of the other θ_j. A diameter containing none of the θ_i is a non-halving diameter and has one open half circle of C with $2n + 1$ points and the other with $2n + 2$ points. The vector in (4.3) will be used to encode the set of lines $S = \{\ell_1, \ldots, \ell_m\}$, where ℓ_i is tangent to C at the point with argument θ_i. The maximum depth cell in $A(S)$ is the regular convex $m-$gon $\mathcal{P}$ that inscribes C and has tangency points at the θ_i. For any x in $\mathcal{P}$ every directional depth will be either $2n + 1$ or $2n+2$, depending on whether or not the direction is orthogonal to a halving diameter, or if it is orthogonal to a non-halving diameter and points to the larger half circle. Therefore $\delta(A(S)) = 2n + 1 = \lfloor m/2 \rfloor$.



The set of inputs to T that we consider are in I, where $z = (z_1, \ldots, z_m) \in I$ if all $z_i \in [0, 2\pi]$ and $z_j = \theta_j,\ j \equiv 1, 2, 3 \bmod 4\}$. Let π and ρ be two different permutations of the integers $4, 8, \ldots, 4n$, and define the input $\theta_\pi \in I$ by

$$(\theta_1, \theta_2, \theta_3, \theta_{\pi_1}, \ldots, \theta_{4n-1}, \theta_{\pi_n}, \theta_{4n+1}, \theta_{4n+2}, \theta_{4n+3})$$

and input $\theta_\rho \in I$ by

$$(\theta_1, \theta_2, \theta_3, \theta_{\rho_1}, \ldots, \theta_{4n-1}, \theta_{\rho_n}, \theta_{4n+1}, \theta_{4n+2}, \theta_{4n+3});$$

the permutations π and ρ point to every fourth entry. Both θ_π and θ_ρ are in $Y \cap I$, as they describe the m lines in S in different order. We argue that they are in different components.

We hold the θ_j fixed for all $j \equiv 1, 2, 3 \bmod 4$, and move each θ_{π_i} continuously on C to the corresponding θ_{ρ_i}, $i = 1, \ldots, n$. This describes a continuous path in I from θ_π to θ_ρ, and every point $z \in I$ on the path describes $m = 4n + 3$ lines tangent to C, with $3n + 3$ of the lines fixed. Since $\pi \neq \rho$, some θ_{4j}

must "cross" (i.e., reverse its radial ordering) a neighbor, θ_{4j-1} or θ_{4j+1} (see figure). Write $k = 4j + 2n + 1 \bmod (m+1)$; θ_k and θ_{k+1} are the neighbors of the diameter through θ_{4j}. If θ_{4j} crosses a neighbor, it first will cross the diameter through θ_k or the diameter through θ_{k+1}. When this occurs that diameter will have $2n$ points on one side and $2n+2$ on the other, and the m lines that are described by this input $z \in I$ will have $\delta(A(S)) = 2n$. Since we have a point on a path from θ_π to θ_ρ that is not in $Y \cap I$, θ_π and θ_ρ must be in different components of $Y \cap I$. This accounts for at least $n!$ different components and since $m = 4n + 3$, we get a lower bound of $\Omega(m \log m)$ on the depth of T. ■

The above argument also establishes an $\Omega(m \log m)$ lower bound for computing the hyperplane depth of a point x. It would be interesting to know if the same lower bound applies to seemingly easier questions. One is the problem of reporting a point x, guaranteed to be in a maximum depth cell of $A(S)$. Another is the problem of reporting the above/below relation of each line with respect to an unknown point x which is in a maximum depth cell.

References

[1] N. Amenta, M. Bern, D. Eppstein, and S.-H. Teng. Regression depth and Center Points. *Discrete and Comp. Geom.* (to appear).

[2] B. Chazelle. Geometric Order statistics. *Proc ACM Geometry*, 1988.

[3] R. Cole. Slowing Down Sorting Networks to Obtain Faster Sorting Algorithms. *J. ACM 34(1)*, 200-208, (1987).

[4] R. Cole, J. Salowe, W. Steiger, and E. Szemerédi. An Optimal Time Algorithm for Slope Selection, *SIAM J. Comp. 18*, 792-810 (1989).

[5] R. Cole, M. Sharir, and K. Yap. On k-Hulls and Related Problems. *Siam J. Comput. 16(1)*,61-77, (1987).

[6] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Berlin, 1987.

[7] J. Gill, W. Steiger, and A. Wigderson. Geometric Medians, *Discrete Math. 108*, 37-51. (1992),

[8] S. Jadhav and A. Mukhopadhyay. Computing a Centerpoint of a Finite Planar Set of Points in Linear Time. *Discrete and Comp. Geom. 12*, 291-312. (1994).

[9] M. van Kreveld, J. Mitchell, P. Rousseeuw, M. Sharir, J. Snoeyink, and B. Speckmann. Efficient Algorithms for Maximum Regression Depth. *Discrete and Comp. Geom.* (to appear).

[10] J. Matoušek. Computing the Center of a Planar Point Set. *Discrete and Computational Geometry: Papers from the DIMACS Special Year* Amer. Math. Soc., J.E. Goodman, R. Pollack, W. Steiger, Eds, (1992), 221-230.

[11] P. Rousseeuw and M. Hubert. Depth in an Arangement of Hyperplanes. *Discrete and Comp. Geom.* (1999)

[12] P. Rousseeuw and M. Hubert. Regression Depth. *J. Amer. Statist. Assoc.*(to appear in June 1999).

[13] W. Steiger and R. Wenger. Hyperplane Depth and Nested Simplices (extended

abstract) 10^{th} *Canadian Conference on Computational Geometry, McGill University, Montreal* (1998).

[14] John Tukey. "Mathematics and the Picturing of Data". *International Conf. of Mathematicians, Vancouver*, 1971.