Computational Geometry, Data Depth, and Robust Statistics

Eynat Rafalin* Diane Souvaine*

July 2004

Abstract

The field of *Computational Geometry* deals with the systematic study of algorithms and data structures for geometric objects. Concepts in robust statistics and specifically the notion of *data depth* for non-parametric multivariate data analysis are inherently geometric in nature, making them well suited for analysis using computational geometry tools and techniques, thus providing faster algorithms and emphasizing the computational aspects of the problems being studied.

The *point-line duality transform*, a basic computational-geometry technique, is well suited for developing efficient algorithms for problems that involve sets of points and lines. In particular it has been used to find efficient algorithms for problems related to statistical analysis of data sets, such as the two dimensional computation of the LMS regression line [51, 18] and the computation of data depth measures.

The area of data-depth-related research, a collaborative effort of statisticians and computational geometers, is the foundation for ongoing research. This paper provides a history of the computational geometry developments in the area, emphasizing the use of computational geometry techniques like point-line-duality, presents recent results in computation of depth contours, half-space depth and simplicial depth and discusses open problems and directions for future research.

1 Introduction

The field of *Computational Geometry* deals with the systematic study of algorithms and data structures for geometric objects [11]. Computational geometry is commonly related to problems in *robotics*, *geographic information systems* and *CAD/CAM*. Any problem that can be represented using geometric objects and operators, however, can be viewed as a computational geometry problem, including the notion of *data depth* for non-parametric multivariate data analysis, an inherently geometric concept.

A *data depth* (see Section 3) measures how deep (or central) a given point $x \in \mathbb{R}^d$ is relative to F, a probability distribution in $\mathbb{R}^d$ or relative to a given data cloud. Data depth provides useful characterization, visualization and quantification tools for high dimensional real life data sets. The data-depth concept provides *center-outward* orderings of points in Euclidean space of any dimension and leads to a new *non-parametric* multivariate statistical analysis in which *no*

*Department of Computer Science, Tufts University, Medford, MA 02155. {erafalin, dls}@cs.tufts.edu

distributional assumptions are needed. Many depth functions are *robust* and *affine invariant* making them well suited for the study of real life high dimensional data sets that may contain outliers.

In order to use the data-depth concept for analysis of real life data sets the issue of computational efficiency, the time and space required to analyze large data sets, should be considered. Computational geometry techniques such as *the point-line duality* and combinatorial and computational results on arrangements of lines contributed to the development of fast, efficient algorithms for statistical-analysis problems, including the computation of the two-dimensional LMS regression line [51, 18] and efficient computation of data depth measures[1]. Further improvements are still needed, for example for computation of data-depth contours. It is possible that the combined computational-geometry and statistical communities can improve either exact running times or approximations for data-depth contours. This paper presents the achievements that have happened so far, the areas most in need of future improvement and some of the dilemmas preventing future improvement.

The technique of point-line duality and the concept of arrangement of lines and their application to LMS will be sketched in Section 2. Survey and results for data-depth related problems will be reviewed in Section 3 including an examination of the concept of *depth contours* and the affect of the contours' definition on the structure and the complexity of their computation, and computation of half-space depth and simplicial depth. Section 4 will include a summary and directions for future work.

2 Computational Geometry Techniques and Computation of the LMS Regression Line [39]

2.1 The Duality Transform [5, 13, 39]

The structure of a collection of lines is more readily observed than the structure implied by a set of points. This structure can be exploited to create efficient algorithms. A set of points can be transformed into an arrangement of lines using a duality transform that preserves key properties [5]. Many problems based on sets of points have been solved efficiently by first transforming the points into a set of lines, solving a related problem on the set of lines, and converting the answer to a solution for the original problem. The transformation itself is conducted in linear time, and consequently does not affect the total computational complexity of the solution. Different transformations apply to different problems. Here, we will focus on a particular transform suitable for the statistical problems we treat. For more details see [13].

The *duality transform* T (see Figure 1) maps a point $P = (a, b)$ to a line $T(P) : y = ax + b$. We need to define the image of a line $l : y = cx + d$ under transform T consistently. Let us pick points $Q = (q, cq+d)$, $R = (r, cr+d)$ lying on line l. $T(Q) : y = qx + (cq + d)$ and $T(R) : y = rx + (cr + d)$. Both of these lines pass through the point $(-c, d)$. Every other point on l will also be mapped to a line passing through $(-c, d)$. Consequently we say that $T(l) : (-c, d)$.

Note that the slope of l is preserved in the x-coordinate of the image point. If the slope of line l exceeds the slope of line k, then $T(l)$ lies to the left of $T(k)$. The vertical distance of a point $P = (a, b)$ to a line $l : y = cx + d$ equals

[1] See also [47, 39]

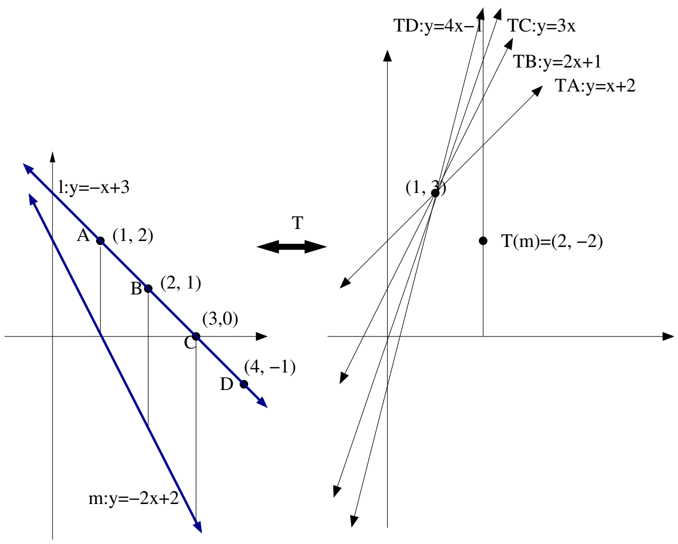


Figure 1: The duality transform T: The set of points and lines on the left is transformed into the set on the right. The points $A = (1, 2), B = (2, 1), C = (3, 0)$ and $D = (4, -1)$ lying on the line $l : y = x + 3$ are mapped to the lines $T(A) : y = x + 2$, $T(B) : y = 2x + 1$, $T(C) : y = 3x$ and $T(D) : y = 4x - 1$, all passing through the point $(1, 3)$, dual to the line l. The points A, B, C, D are located above $m : y = 2x + 2$, as are their dual lines, above point $T(m) : (2, -2)$. In addition, the vertical distance between every point A, B, C, D to the line m is identical to the vertical distance between their dual lines and the dual point $T(m)$.

$b-(ca+d)$. In the dual, the vertical distance from the line $T(p) : y = ax+b$ to the point $T(l) = (-c, d)$ is $(-ac + b) - d$, the same value. Hence, the transformation T preserves slope, vertical distance and the above/below relationship.

For a vertical line $l : x = m$, T maps it to a pencil of parallel lines of slope m. We assume that these parallel line intersect at a point ∞_m (this point is called an *improper point*). For every direction m, there is an improper point ∞_m, associated with it. Conversely, each line $y = mx + a$ which passes through ∞_m gets mapped to a point $(-m, a)$ which lies on the line $x = -m$. Consequently, the image of the improper point ∞_m under T is the vertical line $x = -m$.

The duality transform T can be extended to higher dimensions: in $\mathbb{R}^3$ the dual of the point $P = (a, b, c)$ is the plane $z = ax + by + c$.

2.2 Levels in an Arrangement of Infinite Lines

Arrangements of lines and curves have attracted much attention in the computational geometry community. Their importance is mainly as a structure used to solve a problem derived from a related structure. For example, a problem on a set of points can be solved by dualizing the set into a set of a lines and solving the related problem, as mentioned above. Arrangements have application in domains such as robotics, computer vision and statistics.Examples include the computation of half-space depth contours and robot motion planning [2].

[2] See [22, 16] for a survey on arrangements and related structures.

The complexity of arrangements is important in order to analyze associated algorithms. The maximum combinatorial complexity of an arrangement of lines in $\mathbb{R}^d$ is $O(n^d)$, where the hidden constant depends on d. The arrangement can be constructed in $O(n^d)$ time. If it is simple, its complexity is $\Theta(n^d)$. The complexity of the *zone* of line l, the collection of faces of the arrangement intersected by l is $\Theta(n^{d-1})$ [16, 19]. In the plane, a linear number of cells intersects a single line (although a linear number of edges can create a single cell).

The *kth level* of an arrangement of lines is the set of points that lie on lines and have at most $k-1$ lines above them and at most $n-k$ lines below them (see Figure 4). Consider an arrangement that was created from a set $\mathcal{P}$ of n points, using the duality transform T. If a point $T(p)$, the image of line p, that lies on line $l = T(L)$, is in the kth level, then line p separates $\mathcal{P}$ and has at most $k-1$ points above it and at most $n-k$ points below it. This property will be used in Section 3.3.1, to compute half-space depth contours.

The kth level of an arrangement, and its dual structure, a *k-set*, is a substructure of an arrangement, worthy of independent study. In the plane the complexity of the kth level is bounded from above by $O(nk^{1/3})$ [12] (i.e. a relatively deep level has hyper-linear size) and from below by $n2^{\Omega(\sqrt{\log k})}$ [53]. The gap between the lower and upper bound is a long standing problem in computational geometry. In $\mathbb{R}^3$ the complexity is $O(nk^{3/2})$ [49].The complexity of the at-most-k-level (($\leq k$)-level), the union of points at level j, for $0 \leq j \leq k$, is $\Theta(nk)$ [1]. In the plane the kth level can be constructed in $O(n+f)\log n$ where f is the complexity of the level.

2.3 Least Median of Squares Regression [39]

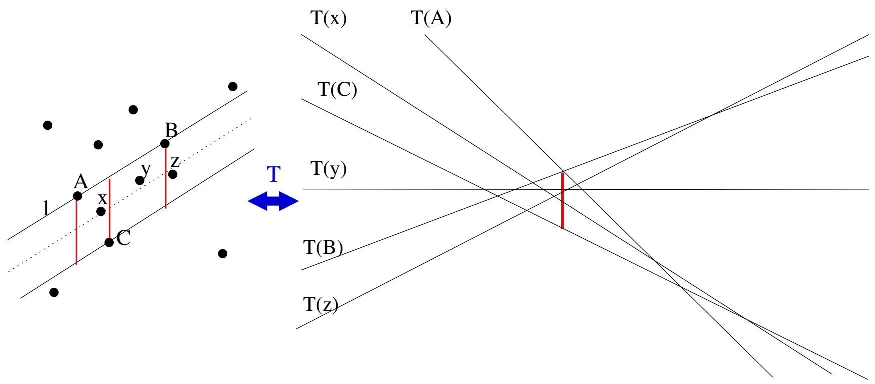


Figure 2: LMS regression: A set of points and its dual arrangement (only some of the lines in the arrangement are drawn), and the LMS slab and regression line. Line l passes through points A and B, and its dual point $T(l)$ is the intersection of lines $T(A)$ and $T(B)$. The slab created from line l and the line parallel to it passing through point C contains 6 points. In the dual, the vertical lines between $T(l)$ and the line $T(C)$ crosses 6 lines.

Consider the problem of fitting a line to a set of data points. The familiar *ordinary least squares (OLS)* method minimizes the sum of the squares of the y-distance between the fitted line and the data points (the *residual*). This method has the disadvantage that a single corrupt point (*outlier*) can significantly perturb the fitted line. The *Least Median of Squares (LMS) Regression line* [41] is the line that minimizes the median of the squares of the residuals. This method has a high breakdown point as up to 50% of the data points can be outliers, without perturbing the fitted line.

It is easy to prove that the problem of finding the line l that minimizes the median residual is equivalent to the problem of finding the slab bounded by a pair of parallel lines of minimum vertical separation that contain $\lceil \frac{n}{2} \rceil$ of the data points. Furthermore, one bounding line L_1 must pass through two points and the other line L_2 through a point whose x-coordinates lies between those of L_1's points. To find the slab, one can check each pair of points (p, q) and find a line l parallel to them, passing through some other point r, such that precisely $\lceil \frac{n}{2} \rceil$ points lie in the closed slab defined by $\overline{pq}$ and l. This can be solved naively using an $O(n^3)$ time algorithm.

Mapping the set of points using the transform T, the above problem becomes one of looking for the intersection point of two lines $T(p)$ and $T(q)$ such that the y-distance between this point and the line $T(r)$, which is $\lceil \frac{n}{2} \rceil$ above or below, is the shortest among all such distances. By sweeping[3] a line through the arrangement dual to the set of points, one can check all intersection points and lines such that the closed vertical segment from the point to the line intersects $\lceil \frac{n}{2} \rceil$ lines and is as short as possible.

Souvaine and Steele [51] proposed an $O(n^2 \log n)$ time algorithm for computing the LMS line in $\mathbb{R}^2$ using the duality concept described above. Their result was improved by Edelsbrunner and Souvaine [18] to an $O(n^2)$ time algorithm, by using a topological line sweep[4]. A practical approximation algorithm for 2 and higher dimensions, using additional computational geometry concepts, was presented by Mount *et al.* [36]. Improved results for the exact and approximation algorithms in any dimension and almost matching lower bounds were recently presented by Erickson *et al.* [20].

3 Data Depth

A *data depth* measures how deep (or central) a given point $x \in \mathbb{R}^d$ is relative to F, a probability distribution in $\mathbb{R}^d$ or relative to a given data cloud. Some examples of data depth functions are *Half-space Depth* [24, 54], *Mahalanobis Depth* [33], *Majority Depth* [50], *Simplicial Depth* [30], *Oja Depth* [38], *Convex Hull Peeling Depth* [3, 15], *L1 Depth* [55] and *Regression Depth* [42].

The data depth concept provides *center-outward* orderings of points in Euclidean space of any dimension and leads to a new *non-parametric* multivariate statistical analysis in which *no distributional assumptions* are needed.

Depth contours are nested contours that enclose regions with increasing depth. They were first introduced by Tukey as a data visualization tool for

[3] Line sweep [4] is a classical technique in computational geometry. The algorithm sweeps a vertical line across an arrangement of objects (points, lines, segments, ..) from left to right, reporting all intersection points, in a series of elementary steps. For an arrangement of n line segments consisting of k intersections points this technique achieves a time complexity of $O((n+k)\log n)$ and requires $O(n)$ space [6]. For an arrangement of n infinite lines that contains $k = \frac{n(n-1)}{2} = O(n^2)$ intersections, vertical line sweep could report all intersection pairs sorted in order of x-coordinate in $\Theta(n^2 \log n)$ time and $O(n)$ space.

[4] In topological line sweep, a topological line, which is monotonic in the y-direction but not necessarily straight, and which intersects each of the n lines in the arrangement exactly once, sweeps the arrangement, in a series of elementary steps, in $O(n^2)$ time and $O(n)$ space [17].

a two dimensional data [54] (half-space depth contours). Since then their use have been expanded for visualizing data sets (e.g. [45]) and quantifying and comparing data sets (e.g. [31]), see also Section 3.2. Depth contours differ according to the underlying depth measure. In addition, several competing definitions for depth contours exist (see Section 3.2), producing distinctive contours (see Figure 3).

In the statistical analysis of datasets, observations that deviate from the main part of the data (*outliers*) can have an undesirable influence on the analysis of the data. Most depth functions are *robust*, yielding reasonable results even if several unannounced outliers occur in the data [32]. For example, in the presence of m outliers only the m outermost depth contours may be corrupted by the outliers, but the inner set of half-space depth contours will maintain its shape [14].

A depth function is said to be *affine invariant* if its coordinate system can change without affecting the relative geometry of the function, i.e. the geometry remains consistent with rotation, scale, or translation of the axes. For such a function, the choice of scale of coordinate system will not affect the relative depth between two points. Thus, for experimental data with several dimensions/variables, even if the relative proportions between the dimensions are not known, the output of the analysis will remain unchanged.

Most depth functions are defined in respect to a probability distribution F, considering $\{X_1, .., X_n\}$ random observations from F. The *finite sample version* of the depth function is obtained by replacing F by F_n, the empirical distribution of the sample $\{X_1, .., X_n\}$. In general, computational geometers study the finite sample case in which sets of points are investigated.

3.1 Data depth as a Characterization, Visualization and Quantification Tool

Tukey [54] was the first to suggest depth contours to indicate the shape of a two dimensional dataset. Later Rousseeuw, Ruts and Tukey [45] proposed the *Bagplot* as a generalization to the univariate *Boxplot*[5]. The two dimensional *Bag* is a convex hull containing half of the observations. The *Fence* is an expansion of the *Bag* by a factor of 3 around the deepest value (this factor is based on simulations). A *loop* indicates the points outside the *Bag* but inside the *Fence*. Every data point outside the *Fence* is flagged as an *outlier*. The resulting graph is called a *Bagplot*. The *Bagplot* is a visualization tool for the location (the depth median), spread (the size of the *Bag*), correlation (the orientation of the *Bag*), skewness (the shape of the *Bag* and the *Loop*) and tails (the points near the boundary of the *loops* and the outliers) of two dimensional data. The *Bagplot* is affine invariant and robust. Liu *et al.* [31] proposed the *Sunburst plot* which is constructed by drawing the contour enclosing 50% of the data points and adding whiskers to the points outside the bag (forming a shape resembling sun with rays).

Liu *et al.* [31] suggested two dimensional depth based graphs to capture and visualize certain properties of high dimensional data sets, such as the *scale curve* to measure scale/dispression, *shrinkage plot* and *fan plot* to measure kurtosis and

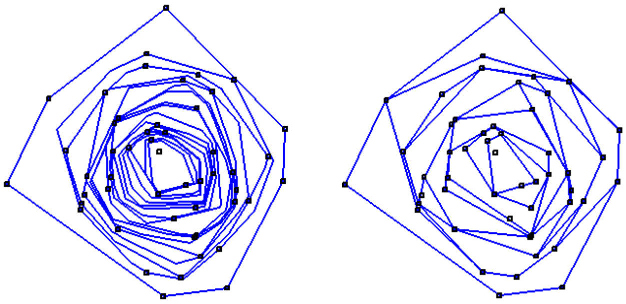
[5]The univariate *Boxplot* consists of a box from the lower quartile to the upper quartile with a crossbar at the median. The *fence* is the region which is the expansion of the *box* by 4 around the median values. Outside the box whiskers are horizontal lines going from the *box* to the most extreme value inside the *fence*. Values outside the Fence are flagged as outliers.

taildness and *DD-plot* to compare the distribution of two data sets.

The depth as a robust classification and cluster analysis tool was proposed by Rousseeuw and Ruts [48], using the concept of measuring the expansion ratio of the α depth contour, instead of the hull enclosing the entire cluster: The robustified cluster analysis partitions the data set into g clusters that minimize the sum of all the volumes of the depth contours that contain α of the points of each cluster (for a certain α). In robustified classification an observation x is classified into set E_j if the difference between the volume of the depth contour D_j^α of E_j that contains α of the data points of the set and the volume of the minimal hull that is the expansion of D_j^α (about its centroid) to include x is minimal of all sets E_i.

3.2 Depth Contours

Several competing definitions for the depth contours in the finite sample case exist, probably due to different translations from the continuous case to the finite sample case. This section provides a framework for viewing the competing definition and analyses their effect on the resulting contours.



(a) **(b)**

Figure 3: Computation of the half-space depth contours according to (a) cover and to (b) rank. The data set consists of 50 points, drawn from a bivariate normal distribution with mean (0,0) and covariance 4 times identity. (a) In the computation according to cover there are 17 contours. Not all the vertices of the contours are original points of the data set. (b) In the computation according to rank the contours were built to enclose 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the data points. The vertices of the contours are the original points of the data set. Several contours share vertices.

Let $D_F(x)$, $x \in \mathbb{R}^d$, be the value of a given depth function for point x with respect to a probability distribution F. For the finite sample case, assume $X = \{X_1, \cdots X_n\}$, and let $D_n(x) = D(x)$ be the depth of $x \in \mathbb{R}^d$ with respect to the data set X.

The set $\{x \in \mathbb{R}^d : D_F(x) = t\}$ is called the **contour of depth** t. We usually refer to the *region enclosed by the contour* of depth t, the set $R_F(t) = \{x \in \mathbb{R}^d : D_F(x) > t\}$.

The α **central region,** C_α $(0 \leq \alpha \leq 1)$ is the smallest region enclosed by depth contours with probability α. Under some restrictions, $C_\alpha = R_F(t_\alpha)$, where $P\{x \in \mathbb{R}^d : D_F(x) \leq t_\alpha\} = \alpha$ (the contour of depth α is the t_αth central region).

We wish to define the sample versions of the *contour of depth* t and the *αth central region*. Several interpretations were suggested and debated by statisticians. We categorize the two main methods as the *cover* approach, most frequently studied by computational geometers, and the *rank* approach, which may produce contours more efficiently. Although lacking some valuable characteristics of the contours produced under the *cover* methods, the *rank*-method depth contours provide a reasonable, and perhaps less expensive, approximation.

3.2.1 Finite Sample Cover contours

The approach based on the notion of **cover** was first introduced by Tukey [54] for describing two dimensional half-space depth contours (e.g. [48, 45, 14, 23]). This approach defines the **sample version of the contour of depth** t as the boundary of the set of points in $\mathbb{R}^d$ with depth $\geq t$. This approach constructs a contour of depth t around all points (not necessarily from the original data set), that are of depth k. To compute the percentage of points of the original data set that are inside the contour one has to count the points of the data set that are inside the contour and compare to the total size of the data set. Rousseeuw *et al.* advocate for this model based on the assumption that the points are random samples from a probability distribution and that the contours should represent the behavior of the distribution and not the specific points.

There are two approaches to construct the **sample version of the α central region**:

- It can be built by enclosing all points that are of depth $D(X_{[\lceil \alpha n \rceil]})$, which is the depth of the $\lceil \alpha n \rceil$ deepest sample point [14, 23]. The contour is constructed by sorting all points of the original set according to their depth (called the *order statistics*), breaking ties at random. A contour that encloses, for example, 75% of the points is created by taking the $\lceil .75n \rceil$ deepest data point $X_{[\lceil .75n \rceil]}$, and constructing the contour of depth $D(X_{[\lceil .75n \rceil]})$. According to this approach the depth contour can be a d dimensional region in $\mathbb{R}^d$, while we look for a $d-1$ dimensional contour. This inconsistency is solved by taking the outer boundary of this region as the contour. The contours converge to contours of the continuous case when $n \rightarrow \infty$ [14].

- Rousseeuw [48, 45, 35] considers the two adjacent contours D_k and D_{k-1} where $D(X_{[\alpha n]}) = k$ (D_k encloses $< \alpha$ of the points, and $D_{k-1} > \alpha$). The αth contour will be constructed *not* by extending the inner contour until it encloses $\lceil \alpha n \rceil$ of the points (as above), *but rather* by interpolating between the coverage of the two contours, according to the percentage of points inside each, using the deepest point as the center. This will create a contour that may have no common point with the data set.

In terms of computational complexity both approaches are similar.

3.2.2 Finite Sample Rank Contours

A different approach, advocated for in [31], is based on **rank**. It defines the **sample αth central region** as the convex hull containing the most central fraction of α sample points. The contour is constructed by sorting all points of the original set according to their depth. A contour that encloses, for example, 75% of the points is created by taking the convex hull around data points $X_{[1]}, \cdots X_{[\lceil .75n \rceil]}$.

3.2.3 Comparison of the Definitions

Under reasonable condition the depth contours for the *continuous* version are equivalent for *cover* and *rank* [31]. However, the contours for the sample version differ significantly. While the *rank* approach maintains the rank order of the sample points the *cover* approach mimics the continuous quantiles. A few differences:

- *Rank* contours can be constructed according to any depth function that calculates the depth of a data point, while contours according to *cover* require one to compute the depth of any other point $x \in \mathbb{R}^d$.

- Both approaches create depth contours that are nested. However, the *cover* method may produce contours that are not connected (if the depth function is not connected, e.g. simplicial depth in [7]). In addition, *rank* contours will always be convex, while the *cover* contours may not be. For the half-space function the *cover* contours are always convex [14].

- The main visual difference between the two approaches are the vertices of the contours. Every data point will be a vertex of (at least one) depth contour. However, the vertices of the *rank* contours will only be data points while the vertices of the *cover* contours can be any point from the data set. A vertex will be common to more then one *rank* contour while only degenerate data sets will create common vertices for *cover* contours.

- Degenerate data sets lead to contours with peculiar properties: Take for example the contour that contains 50% of the data points, but assume a degenerate set that contains more then 50% of the data points on its convex hull. Different depth ordering of the data points of minimal depth (points on the convex hull) will create varying contours according to the *cover* approach, depending on which of the points on the convex hull is chosen to be in the 50% inner points[6]. According to Donoho and Gasko's *cover* approach the 50% contour will be identical to the 100% contour since the depth of the $X_{[\lceil n/2 \rceil]}$ point is identical to the depth of any point on the convex hull. Only Rousseeuw's *cover* approach will create different, non overlapping, contours for every percentile, since it interpolates between the $< 50\%$ contour and the $> 50\%$ contour which is the outer convex hull.

The cover approach was most frequently studied by computational geometers. The rank approach may produce contours more efficiently and provides a reasonable, and perhaps less expensive, approximation

3.3 Half-space Depth and Half-space Depth Contours

Definition 1. The *half-space depth*[7] of a point x relative to a set of points $\mathcal{S} = \{X_1, \ldots, X_n\}$ is the minimum number of points of $\mathcal{S}$ lying in any closed half-space determined by a line through x [24, 54].

Using the duality transform T, the set of point $\mathcal{S}$ maps to an arrangement of lines $T(\mathcal{S})$. The number k of points of $\mathcal{S}$ above (resp. below) a line L through

[6] To remedy this [31] require that equal depth points belong to the same sample pth level contour for some p. However in degenerate cases such as when all the points are on the convex hull of the set, this constructs only one valid contour.

[7] In the literature this is sometimes called *location depth* or *Tukey depth*.

x, equals the number of lines of $T(\mathcal{S})$ above (resp. below) the dual point $T(L)$. The depth of a point x relative to $\mathcal{S}$ is the minimum number of points of $\mathcal{S}$ lying in any closed half-space determined by a line through x (above or below the line through x). Consequently, the depth of a line $T(x)$, dual to the point x, relative to $T(\mathcal{S})$, is the minimal number of lines of $T(\mathcal{S})$ above or below $T(x)$.

3.3.1 Efficient computation of Depth Contours using the Duality Transform [39]

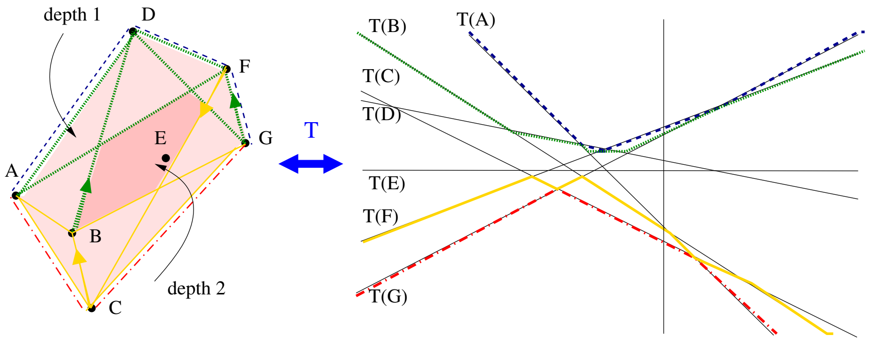


Figure 4: Levels in an arrangement and half-space depth computation: The set of 7 points $A-G$ on the left is transformed using the duality transform T into the arrangement of lines $T(A)-T(G)$ on the right. The levels $1, 2, 6, 7$ are drawn in blue, green, yellow and red, respectively. Any vertical line cutting the kth level will pass through exactly $k-1$ lines of the arrangement above it. The intersection points on the 1st and 7th level are candidate points for the first half-space depth contour (as are the 2nd and 6th levels for the second depth contour). Therefore transforming the intersection points on each level back to the primal plane (as lines) and computing their intersection will result in that depth contour. The 1st and 2nd contours are drawn as grayed areas.

The kth *depth contour* according to the cover approach (see also Section 3.2) for a set of points $\mathcal{S}$ in $\mathbb{R}^d$ is the boundary of the points of $\mathbb{R}^d$ with depth $\geq k$, according to the chosen depth function. The kth depth contours for half-space depth in $\mathbb{R}^2$ can be constructed by taking the intersection of all half-planes containing k points of $\mathcal{S}$. Specifically, the significant half-planes are the ones bounded by lines connecting pairs of points of $\mathcal{S}$. Every line L passing through a pair of points l, m is dualized to an intersection point I of two lines $T(l), T(m)$. The vertical line through I in the dual intersects every line of the arrangement, some above I and some below. The number of lines intersected above (resp. below) I equals the number of points lying above (resp. below) $T^{-1}(I) = L$ in the primal. The contour to which L contributes is the minimum of these two numbers (this is also said to be the *level* of L, see Section 2.2). Examining all intersection points in the arrangement of dual lines will produce a list of half-planes/intersections for each potential contour.

The duality concept can be easily extended to higher dimensions. For example, in $\mathbb{R}^3$ the half-space depth of a point x is the minimum number of points of a given set $\mathcal{S}$ lying in any closed half-space bounded by a hyperplane (instead of a line, as in $\mathbb{R}^2$) through x.

Miller *et al.* [35] presented an $O(n^2)$ time algorithm for computing all depth contours for a planar data set, using the idea sketched above. The authors used *topological sweep* (see Section 2.3 and [17]) to examine all intersection points and to report the level of each intersection point in optimal time and space.

Other half-space-depth related computational results include the computation of the depth of a single point in $O(n \log n)$ time [48], computation of the exact and approximated deepest point relative to a data set (the *Tukey median*) [48, 44, 46, 52, 56], computation of *centerpoints*, which are points of depth $\geq n/(d+1)$ [9, 25, 34, 37] and results on the computation of depth contours, both theoretical [10, 34, 29] and practical [48, 26, 35, 40, 28].

3.4 Simplicial Depth [30, 7, 39]

Simplicial depth was introduced by Liu in 1990 [30]:

Definition 2. The *simplicial depth* of a point x with respect to a data set $\mathcal{S} = \{X_1, \ldots, X_n\}$ is the fraction of the closed simplicies formed by $d+1$ points of $\mathcal{S}$ containing x, where I is the indicator function: $SD_{Liu}(\mathcal{S}; x) = \binom{n}{d+1}^{-1} \sum I_{(x \in S[X_{i_1}, \ldots X_{i_{d+1}}])}$ (see Figure 5(a)).

A simplex in $\mathbb{R}^d$ is the boundary of the region defined by $d+1$ vertices in general position, and represents the simplest possible polytope in any dimension. In one dimension, a simplex is simply a line segment; in 2 dimensions, it is a triangle; and in 3 dimensions, a tetrahedron. The simplicial depth is computationally more difficult in the finite sample case than some other depth measures.

3.4.1 A Revised Definition [7]

Several problems arise in the finite sample case of simplicial depth under Liu's definition. As suggested by Zuo and Serfling [57], a depth function should attain maximum value at the center (*the maximality property*) and as a point x moves away from the *deepest point* along any fixed ray through the center, the depth at x should decrease monotonically (*monotonicity property*). However, the simplicial depth function for the finite sample case fails to satisfy these properties [57, 7] (see e.g. Figure 5(b)). In addition, the depth of points on facets causes discontinuities in the depth function: The depth of all points on the boundary of a cell is at least the depth of a point on the interior of the cell. In most cases the depth values on the boundaries can be higher than the depth in each of the adjacent cells (see e.g. Figure 5(b)).

Burr *et al.* [7] proposed a modification to the definition that corrects the irregularity at boundaries of simplicies by making the depth of a point on the boundary between two cells the *average* of the depth of the two cells and fixes the counterexamples provided by Zuo and Serfling [57]:

Definition 3. Given a data set $\mathcal{S} = \{X_1, \ldots, X_n\}$ in $\mathbb{R}^d$, the *simplicial depth* of a point x is the average of the fraction of closed simplicies containing x and the fraction of open simplicies containing x:

$$SD_{BRS}(\mathcal{S}; x) = \frac{1}{2}\binom{n}{d+1}^{-1} \left(\sum I_{(x \in S[X_{i_1}, \ldots X_{i_{d+1}}])} + I_{(x \in int(S[X_{i_1}, \ldots X_{i_{d+1}}]))}\right) \tag{3.1}$$

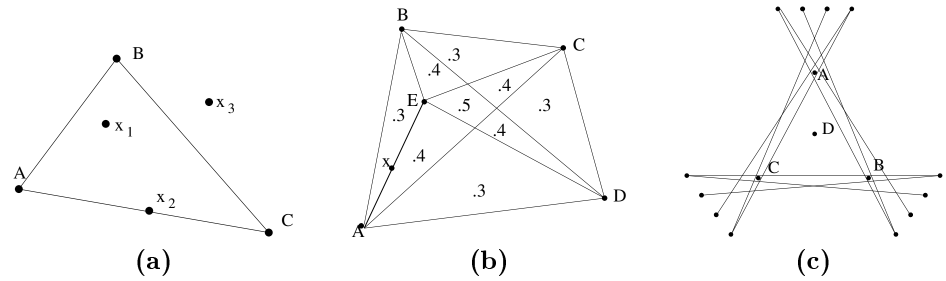


Figure 5: (a) Computation of simplicial Depth, according to Liu and according to Burr *et al.*. $SD_{Liu}(x_1) = SD_{BRS}(x_1) = 1$, $SD_{Liu}(x_2) = 1$, $SD_{BRS}(x_2) = .5$. $SD_{Liu}(x_3) = SD_{BRS}(x_3) = 0$. (b) Problems with simplicial depth: The total number of simplicies is $\binom{5}{3} = 10$. The depth of the open regions is drawn on the figure. $SD_{Liu}(p) = \binom{4}{2}/10 = .6$, for $p = \{A, B, C, D\}$, while $SD_{Liu}(E) = .8$. For a point x on $\overline{AE}$ $SD_{Liu}(x) = .5$, violating the monotonicity property and causing discontinuities in the depth function at edges. According to the revised definition $SD_{BRS}(p) = .3$, for $p = \{A, B, C, D\}$, $SD_{BRS}(E) = .5$ and $SD_{BRS}(x) = .35$. (c) A problem with the revised definition. The data points A,B, and C all have depth $\frac{587}{1120}$ and the data point D, which is at the unique center of the data set has depth $\frac{355}{1120}$. For clarity reasons not all cells are drawn.

where *int* refers to the open relative interior[8] of $S[X_{i_1}, \ldots, X_{i_{d+1}}]$. Equivalently, this could be formulated as: $SD_{BRS}(\mathcal{S}; x) = \rho(\mathcal{S}, x) + \frac{1}{2}\sigma(\mathcal{S}, x)$ where $\rho(\mathcal{S}, x)$ is the number of simplicies with data points as vertices which contain x in their open interior, and $\sigma(\mathcal{S}, x)$ is the number of simplicies with data points as vertices which contain x in their boundary.

The revised definition reduces to the original definition for continuous distributions and for points lying in the interior of cells, and it maintains the ranking order of data points. In addition, it can be calculated using the existing algorithms with slight modifications [7]. However, it does not achieve all desired properties in the sample case. Figure 5(c) shows an example where the data set has a unique center, D, but it neither attains maximality at the center, nor does it have monotonicity relative to the deepest point.

3.4.2 Overview of Related Results

Other relevant results include computing the simplicial depth for a single point in $\mathbb{R}^2$ in $O(n \log n)$ time and in $\mathbb{R}^3$ and $\mathbb{R}^4$ [21, 27, 43, 8], computing the simplicial depth for all points in $\mathbb{R}^2$ in $O(n^2)$ using the duality transform [21, 27] and finding the simplicial median in $\mathbb{R}^2$ [2, 7].

4 Summary

This paper first surveyed computational geometry methods and techniques and their use for solving problems related to statistical analysis using robust statistics (LMS regression line and data depth functions) and then focused more specifically on data depth. Research has deepened understanding about which

[8] See Edelsbrunner [16], page 401.

depth measure is more appropriate for a certain application or a data set and what attributed are best preserved by each measure. However, more work is needed. It is relatively known how depth contours apply in the case of half-space depth. We are currently investigating simplicial depth and regression depth contours.

The use of computational geometry techniques on data-depth measures have produced more efficient algorithms and implementations with significant speedup. However, most algorithms and their implementations are not practical for real-life data sets as they are designed for low-dimensional data sets and in some cases although the asymptotic running times are efficient, in practice, the hidden constants make implementations infeasible. Exact computation of most depth functions will probably never be efficient enough in high dimensions, as the complexity of the algorithm is in most cases exponential in the dimension of the data. The solution probably lies in approximation algorithms, that can compute depth functions and depth contours in high dimensions efficiently, with provable error bounds. Another important direction is that of dynamically computing depth contours, where the set of depth contours is updated as data points are inserted to and deleted from the data set.

References

[1] Noga Alon and E. Győri. The number of small semispaces of a finite set of points in the plane. *J. Combin. Theory Ser. A*, 41(1):154–157, 1986.

[2] G. Aloupis, S. Langerman, M. Soss, and G. Toussaint. Algorithms for bivariate medians and a Fermat-Torricelli problem for lines. *Comp. Geom. Theory and Appl.*, 26(1):69–79, 2003.

[3] V. Barnett. The ordering of multivariate data. *J. Roy. Statist. Soc. Ser. A*, 139(3):318–355, 1976.

[4] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28(9):643–647, September 1979.

[5] K. Q. Brown. *Geometric transforms for fast geometric algorithms.* Ph.D. thesis, Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, 1980. Report CMU-CS-80-101.

[6] K. Q. Brown. Comments on "Algorithms for reporting and counting geometric intersections". *IEEE Trans. Comput.*, C-30:147–148, 1981.

[7] M. Burr, E. Rafalin, and D. L. Souvaine. Simplicial depth: An improved definition, analysis, and efficiency for the discrete case. DIMACS technical report 2003-28, Rutgers University–New Brunswick, 2003. appeared in Proc. 16th Canadian Conference on Computational Geometry.

[8] A. Y. Cheng and M. Ouyang. On algorithms for simplicial depth. In *Proc. 13th Canadian Conference on Computational Geometry*, pages 53–56, 2001.

[9] K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterated Radon points. *Int. J. Comp. Geom. Appl.*, 6(3):357–377, 1996.

[10] R. Cole, M. Sharir, and C. K. Yap. On k-hulls and related problems. *SIAM Journal on Computing*, 15(1):61–77, 1987.

[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 1997.

[12] T. K. Dey. Improved bounds for planar k-sets and related problems. *Discrete Comput. Geom.*, 19(3, Special Issue):373–382, 1998. Dedicated to the memory of Paul Erdős.

[13] D. P Dobkin and D. L. Souvaine. Computational geometry - a user's guide. In J. T. Schwartz and C. K. Yap, editors, *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*, pages 43–93. Lawrence Erlbaum Associates, 1987.

[14] D. L. Donoho and M. Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.*, 20(4):1803–1827, 1992.

[15] W. Eddy. Convex hull peeling. In H. Caussinus, editor, *COMPSTAT*, pages 42–47. Physica-Verlag, Wien, 1982.

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

[17] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989.

[18] H. Edelsbrunner and D. L. Souvaine. Computing median-of-squares regression lines and guided topological sweep. *J. Amer. Statist. Assoc.*, 85:115–119, 1990.

[19] Herbert Edelsbrunner, Raimund Seidel, and Micha Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22(2):418–429, 1993.

[20] J. Erickson, S. Har-Peled, and D. Mount. On the least median squares problem. In *Proc. 25th Annual ACM Symposium on Computational Geometry*, pages 273–279, 2004.

[21] J. Gil, W. Steiger, and A. Wigderson. Geometric medians. *Discrete Math.*, 108(1-3):37–51, 1992. Topological, algebraical and combinatorial structures. Frolík's memorial volume.

[22] D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 529–562. CRC Press LLC, Boca Raton, FL, 2004.

[23] X. He and G. Wang. Convergence of depth contours for multivariate datasets. *Ann. Statist.*, 25(2):495–504, 1997.

[24] J. Hodges. A bivariate sign test. *The Annals of Mathematical Statistics*, 26:523–527, 1955.

[25] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete & Computational Geometry*, 12(3):291–312, 1994.

[26] T. Johnson, I. Kwok, and R. Ng. Fast computation of 2-dimensional depth contours. In *Proc. 4th International Conference on Knowledge Discovery and Data Mining*, pages 224–228, 1998.

[27] S. Khuller and J. S. B. Mitchell. On a triangle counting problem. *Inform. Process. Lett.*, 33(6):319–321, 1990.

[28] S. Krishnan, N. H. Mustafa, and S. Venkatasubramanian. Hardware-assisted computation of depth contours. In *13th ACM-SIAM Symposium on Discrete Algorithms*, 2002.

[29] S. Langerman and W. Steiger. Optimization in arrangements. In *Proceedings of the 20th International Symposium on Theoretical Aspects of Computer Science (STACS 2003)*, 2003.

[30] R. Liu. On a notion of data depth based on random simplices. *The Annals of Statistics*, 18:405–414, 1990.

[31] R. Liu, J. Parelius, and K. Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference (with discussions). *The Annals of Statistics*, 27:783–858, 1999.

[32] G. S. Maddala and C. R. Rao, editors. *Robust inference*, volume 15 of *Handbook of Statistics*. North-Holland Publishing Co., Amsterdam, 1997.

[33] P. C. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Academy India*, volume 12, pages 49–55, 1936.

[34] J. Matoušek. Computing the center of planar point sets. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 6:221–230, 1991.

[35] K. Miller, S. Ramaswami, P. Rousseeuw, T. Sellarés, D. Souvaine, I. Streinu, and A. Struyf. Efficient computation of location depth contours by methods of combinatorial geometry. *Statistics and Computing*, 13(2):153–162, 2003.

[36] D. M. Mount, N. S. Netanyahu, K. R. Romanik, R. Silverman, and A. Y. Yu. A practical approximation algorithm for the LMS line estimator. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 473–482, January 1997.

[37] N. Naor and M. Sharir. Computing a point in the center of a point set in three dimensions. In *Proc. 2nd Canadian Conference on Computational Geometry*, pages 10–13, 1990.

[38] H. Oja. Descriptive statistics for multivariate distributions. *Statistics and Probability Letters*, 1:327–332, 1983.

[39] E. Rafalin and D. Souvaine. Computational geometry and statistical depth measures. In M. Hubert, G. Pison, A. Struyf, and S. Van Aelst, editors, *Theory and Applications of Recent Robust Methods*, Statistics for Industry and Technology. Birkhauser, Basel, 2004.

[40] V. Rosta and K. Fukuda. Exact parallel algorithms for the location depth and the maximum feasible subsystem problems. In C.A. Floudas and P.M. Pardalos, editors, *Frontiers In Global Optimization*. Kluwer Academic Publishers, 2003.

[41] P. J. Rousseeuw. Least median of squares regression. *J. Amer. Statist. Assoc.*, 79(388):871–880, December 1984.

[42] P. J. Rousseeuw and M. Hubert. Regression depth. *J. Amer. Statist. Assoc.*, 94:388–433 (with discussion), 1999.

[43] P. J. Rousseeuw and I. Ruts. Bivariate location depth. *Applied Statistics-Journal of the Royal Statistical Society Series C*, 45(4):516–526, 1996.

[44] P. J. Rousseeuw and I. Ruts. Constructing the bivariate Tukey median. *Statistica Sinica*, 8:827–839, 1998.

[45] P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The bagplot: A bivariate boxplot. *The American Statistician*, 53:382–387, 1999.

[46] P. J. Rousseeuw and A. Struyf. Computing location depth and regression depth in higher dimensions. *Statistics and Computing*, 8:193–203, 1998.

[47] Peter J. Rousseeuw and Anja Struyf. Computation of robust statistics: Depth, median, and related problems. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, pages 1279–1292. Chapman and Hall / CRC, 2004.

[48] I. Ruts and P. J. Rousseeuw. Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, 23:153–168, 1996.

[49] Micha Sharir, Shakhar Smorodinsky, and Gábor Tardos. An improved bound for k-sets in three dimensions. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry (Hong Kong, 2000)*, pages 43–49 (electronic), New York, 2000. ACM.

[50] K. Singh. On the majority data depth. Technical report, Department of Statistics, Rutgers University, 1993.

[51] D. L. Souvaine and J. M. Steele. Time- and space- efficient algorithms for least median of squares regression. *J. Amer. Statist. Assoc.*, 82:794–801, 1987.

[52] A. Struyf and P. Rousseeuw. High-dimensional computation of the deepest location. *Computational Statistics & Data Analysis*, 34:415–426, 2000.

[53] G. Tóth. Point sets with many k-sets. *Discrete Comput. Geom.*, 26(2):187–194, 2001. ACM Symposium on Computational Geometry (Hong Kong, 2000).

[54] J. W. Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians (Vancouver, B. C., 1974), Vol. 2*, pages 523–531. Canad. Math. Congress, Montreal, Que., 1975.

[55] Yehuda Vardi and Cun-Hui Zhang. The multivariate L_1-median and associated data depth. *Proc. Natl. Acad. Sci. USA*, 97(4):1423–1426 (electronic), 2000.

[56] K. Verbarg. Approximate center points in dense point sets. *Inform. Process. Lett.*, 61(5):271–278, 1997.

[57] Y. Zuo and R. Serfling. General notions of statistical depth function. *Ann. Statist.*, 28(2):461–482, 2000.