Curve Stabbing Depth: Data Depth for Plane Curves*

Stephane Durocher† Spencer Szabados†

Abstract

Measures of data depth have been studied extensively for point data. Motivated by recent work on analysis, clustering, and identifying representative elements in sets of trajectories, we introduce *curve stabbing depth* to quantify how deeply a given curve Q is located relative to a given set $\mathcal{C}$ of curves in $\mathbb{R}^2$. Curve stabbing depth evaluates the average number of elements of $\mathcal{C}$ stabbed by rays rooted on Q. We describe an $O(nm^2 + n^2 m \log^2 m)$-time algorithm for computing curve stabbing depth when Q is an m-vertex polyline and $\mathcal{C}$ is a set of n polylines, each with $O(m)$ vertices.

1 Introduction

Processes that generate functional or curve data are becoming increasingly common within various domains, including medicine (e.g., ECG signals [16] and analysis of nerve fibres in brain scans [11]), GIS techniques for generating and processing positional trajectory data (e.g., tracking migratory animal paths [4], air traffic control [8], and clustering of motion capture data [12]), and in the food industry (e.g., classification of nutritional information via spectrometric data [13]). In this paper, we consider depth measures for curve data.

Traditional depth measures are defined on multidimensional point data and seek to quantify the centrality or the outlyingness of a given object relative to a set of objects or to a sample population. Common depth measures include simplicial depth [18], Tukey (half-space) depth [23], Oja depth [20], convex hull peeling depth [3], and regression depth [21]. See [19] and [22] for further discussion on depth measures for multivariate point data. Previous work exists defining depth measures for sets of functions and functional data [13, 10, 16, 9], often with a focus on classification. Despite the fact that curves can be expressed as functions, depth measures for functions typically do not generalize to curves, as they are often sensitive to the specific parameterization and most are restricted to functions whose range is $\mathbb{R}$, which can only represent x-monotone curves.

New methods are required for efficient analysis of trajectory and curve data. Recent work has examined identifying representative elements [4] and clustering in a set of trajectories [5, 12]. In this work, we introduce curve stabbing depth, a new depth measure defined in terms of stabbing rays to quantify the degree to which a given curve is nested within a given set of curves.

Our main contributions are:

- In Section 2, we define *curve stabbing depth*, a new depth measure for curves in $\mathbb{R}^2$, and we describe a general approach for evaluating the curve stabbing depth of a given curve Q relative to a set $\mathcal{C}$ of curves in $\mathbb{R}^2$.
- In Section 3, we present an $O(nm^2 + n^2 m \log^2 m)$-time algorithm for computing the curve stabbing depth of a given m-vertex polyline Q relative to a set $\mathcal{P}$ of n polylines in $\mathbb{R}^2$, each with $O(m)$ vertices.
- In Section 4, we discuss properties of a deepest curve (depth median) for curve stabbing depth, discuss the consistency of generalizations to higher dimensions, and outline possible directions for future research.

2 Definitions

Definition 1 (Plane Curve) *A* plane curve *is a continuous function $Q : [0, 1] \to \mathbb{R}^2$.*

Definition 2 (Polyline) *A* polyline *(polygonal chain) is a piecewise-linear curve consisting of the line segments $\overline{p_1p_2}, \overline{p_2p_3}, \ldots, \overline{p_{m-1}p_m}$ determined by the sequence of points $(p_1, p_2 \ldots, p_m)$ in $\mathbb{R}^2$.*

Definition 3 (Stabbing Number) *Given a ray $\overrightarrow{q_\theta}$ rooted at a point q in $\mathbb{R}^2$ that forms an angle θ with the x-axis, the* stabbing number *of $\overrightarrow{q_\theta}$ relative to a set $\mathcal{C}$ of plane curves, denoted $\mathrm{stab}_{\mathcal{C}}(\overrightarrow{q_\theta})$, is the number of elements in $\mathcal{C}$ intersected by $\overrightarrow{q_\theta}$.*

Definition 4 (Curve Stabbing Depth) *Given a plane curve Q and a set $\mathcal{C}$ of plane curves, the* curve stabbing depth *of Q relative to $\mathcal{C}$, denoted $D(Q, \mathcal{C})$, is*

$$\frac{1}{\pi L(Q)} \int_{q \in Q} \int_0^{\pi} \min\{\mathrm{stab}_{\mathcal{C}}(\overrightarrow{q_\theta}), \mathrm{stab}_{\mathcal{C}}(\overrightarrow{q_{\theta+\pi}})\}\, d\theta\, ds, \quad \text{(D.c)}$$

where $L(Q) = \int_{q \in Q} ds$ denotes the arc length of Q.

*This work is funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

†University of Manitoba, Winnipeg, Canada, stephane.durocher@umanitoba.ca, szabados@myumanitoba.ca

As $L(Q)$ approaches zero (the curve Q becomes a point q), the value of (D.c) approaches

$$\frac{1}{\pi}\int_0^{\pi} \min\{\text{stab}_{\mathcal{C}}(\overrightarrow{q\theta}), \text{stab}_{\mathcal{C}}(\overrightarrow{q\theta+\pi})\}\, d\theta. \qquad \text{(D.p)}$$

Curve stabbing depth corresponds to the average depth of points $q \in Q$. The depth of a point q relative to $\mathcal{C}$, given by (D.p), is the average stabbing number in all directions θ around q, where for each θ, either the stabbing number of the ray $\overrightarrow{q\theta}$ or its reflection $\overrightarrow{q\theta+\pi}$ is applied, generalizing the one-dimensional notion of depth that counts the lesser of the number of elements less than vs. greater than the query point (outward rank).

As a ray $\overrightarrow{q\theta}$ rotates about a point q, $\text{stab}_{\mathcal{C}}(\overrightarrow{q\theta})$ partitions the range $\theta \in [0, \pi)$ into intervals, such that for all values θ in a given interval, $\overrightarrow{q\theta}$ intersects the same subset of $\mathcal{C}$. These intervals partition the plane around q into *wedges*. We generalize this notion and define the wedges determined by a point q relative to a set $\mathcal{C}$ of curves.

Definition 5 (Wedge) *The* wedge *of the curve C relative to the point q is the region determined by all rays rooted at q that intersect C:*

$$w(q, C) = \bigcup_{\substack{\overrightarrow{q\theta} \cap C \neq \varnothing \\ \theta \in [0, 2\pi)}} \overrightarrow{q\theta}.$$

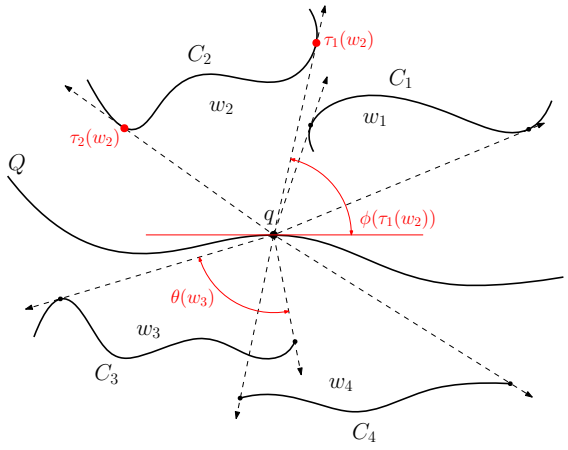


Figure 1: The set of wedges w_1, w_2, w_3, and w_4 induced by curves C_1, C_2, C_3, and C_4 rooted at the point q on the curve Q. Moving counterclockwise around q, the positive angle between $\tau_1(w_2)$ with the horizontal is indicated by $\phi(\tau_1(w_2))$, the tangent points of w_2 are labelled $\tau_1(w_2)$ and $\tau_2(w_2)$, and the internal angle of w_3 is highlighted by $\theta(w_3)$.

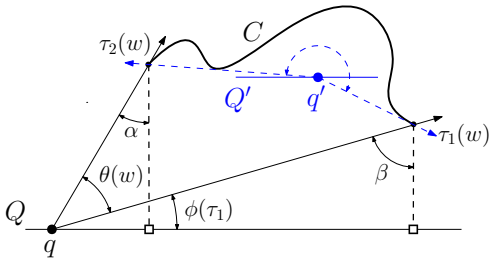
Definition 6 (Tangent Points) *When $C \cup \{q\}$ is in general position in $\mathbb{R}^2$, the* tangent points *of the wedge $w = w(q, C)$, denoted $\tau(w) = \{\tau_1, \tau_2\}$, are those points of C incident with the boundary of w; i.e., $\tau(w) = \partial w \cap C$, where ∂w denotes the boundary of w. (If C is a curve for which all rays from q intersect, the tangents points of $w(q, C)$ are taken to be coincident on C, with an internal wedge angle of 2π radians.) $\tau_1(w)$ denotes the tangent point that is the most clockwise of the two around q. The angles between the horizontal and each tangent point of w are denoted by $\phi(\tau_1(w))$ and $\phi(\tau_2(w))$, with $\theta(w)$ denoting the interior angle of w.*

See Figures 1 and 2. The sequence of wedges determines an ordering of the curves stabbed about a given point q. A ray $\overrightarrow{q\theta}$ always stabs the associated curve C as $\overrightarrow{q\theta}$ sweeps through the wedge determined by the extreme points of C. For a given set $\mathcal{C}$ of curves and associated wedges $\mathcal{W}_{\mathcal{C}}$ rooted at a common point q,

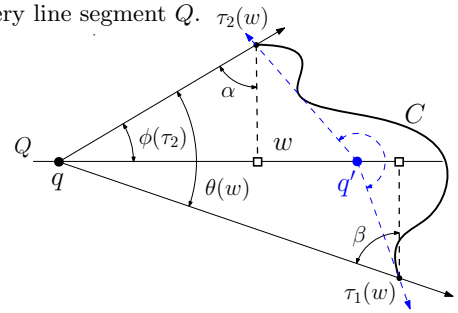
$$\text{stab}_{\mathcal{C}}(\overrightarrow{q\theta}) = |\{w \in \mathcal{W}_{\mathcal{C}} \mid \theta \in [\phi(\tau_1(w)), \phi(\tau_2(w))]\}|. \qquad (1)$$

That is, $\text{stab}_{\mathcal{C}}(\overrightarrow{q\theta})$ is the number of wedges that contain the ray $\overrightarrow{q\theta}$, where each wedge is associated with a curve in $\mathcal{C}$. See Figure 1.

Our algorithm for computing curve stabbing depth requires calculating the interior angle $\theta(w)$ of each wedge w, which we now describe. We consider two cases for the relative positions of a given query line segment Q, a curve C, and the wedge $w(q, C)$ rooted at a point q: (1) when $q \notin \text{CH}(C)$, where $\text{CH}(C)$ denotes the convex hull of C, i.e., Q does not pass through the interior of C, and (2) when $q \in \text{CH}(C)$. When points and curves are in general position, C cannot coincide with a bounding edge of w. See Figure 2.



(a) A curve C existing entirely above (below) the query line segment Q.



(b) A curve C crossing through the query line segment Q.

Figure 2: Two ways a query line segment C and a wedge rooted at a point on C can be arranged under general position. Case 1 is drawn in black while Case 2 is outlined in blue.

In Case 1, when C lies entirely above or below Q the angles formed between the tangent points, root, and horizontal can be evaluated as

$$\theta(w) + \phi(\tau_1) = \frac{\pi}{2} - \alpha \quad = \frac{\pi}{2} - \tan^{-1}\left(\left|\frac{q_x - \tau_2(w)_x}{\tau_2(w)_y - q_y}\right|\right),$$
$$\phi(\tau_1) = \frac{\pi}{2} - \beta \quad = \frac{\pi}{2} - \tan^{-1}\left(\left|\frac{q_x - \tau_1(w)_x}{\tau_1(w)_y - q_y}\right|\right).$$

Where the interior angle of w is found to be

$$\theta(w) = \tan^{-1}\left(\frac{q_x - \tau_2(w)_x}{\tau_2(w)_y - q_y}\right) - \tan^{-1}\left(\frac{q_x - \tau_1(w)_x}{\tau_1(w)_y - q_y}\right). \tag{A.1}$$

This can be done rather than evaluating distinct cases due to the order in which the signs of each inverse tangent change while q transitions past each dropped perpendicular. When C crosses in front of Q, as illustrated in Figure 2b, we calculate

$$\theta(w) = \pi - \left|\tan^{-1}\left(\frac{q_x - \tau_2(w)_x}{\tau_2(w)_y - q_y}\right) + \tan^{-1}\left(\frac{q_x - \tau_1(w)_x}{\tau_1(w)_y - q_y}\right)\right|. \tag{A.2}$$

Once q enters $\mathrm{CH}(C)$, we transition to Case 2, in which the calculations are similar to those of Case 1, except for modifications needed to account for taking an angle greater than π radians, as shown in Figure 2b in blue. Every case considered by our algorithm reduces to Case 1 or Case 2. We sometimes limit discussion to instances of Case 1 depicted in Figure 2a to simplify the presentation; our results apply to all cases.

Definition 7 (Circular Partition) *The* circular partition *induced by the set of wedges* $\mathcal{W}_\mathcal{C} = \{w_1, w_2, \ldots, w_n\}$ *rooted at a common point* q *is the sequence* $0 = \theta_0 < \theta_1 < \cdots < \theta_{4n} < 2\pi$ *of angles, corresponding to the ordered sequence of bounding edges of wedges in* $\mathcal{W}_\mathcal{C}$*; i.e., it is the ordered sequence of values in* $\{\theta_i \mid \theta_i \in \{\phi(\tau_j), \phi(\tau_j) + \pi \bmod 2\pi\}, \tau_j \in \tau(w), w \in \mathcal{W}_\mathcal{C}\}$. *Denote this sequence by* $\sigma(\mathcal{W}_\mathcal{C}) = (\theta_0, \theta_1, \ldots, \theta_{4n})$.

See Figure 3. Applying Equation (1) to Definition 7, we arrive at the following observation:

Observation 1 *Given a set* $\mathcal{W}_\mathcal{C}$ *of wedges and induced partition* $\sigma(\mathcal{W}_\mathcal{C}) = (\theta_0, \theta_1, \ldots, \theta_{4n})$ *for a given point* q *and set* $\mathcal{C}$ *of curves, for every* $i \in \{1, \ldots, 4n-1\}$ *and every* $\phi_1, \phi_2 \in (\theta_i, \theta_{i+1})$*, the set of curves in* $\mathcal{C}$ *intersected by* $\overrightarrow{q_{\phi_1}}$ *is the same as that intersected by* $\overrightarrow{q_{\phi_2}}$.

Observation 1 remains true when the point q at the root of the wedges moves within a bounded neighbourhood: given a curve Q and a set $\mathcal{C}$ of curves, for each point q on Q, the relative ordering of wedge boundaries in the circular partition of q remains unchanged when q moves along some interval of Q. By partitioning Q into such cyclically invariant segments, this property allows us to calculate the curve stabbing depth of Q relative to $\mathcal{C}$ discretely. Formally:

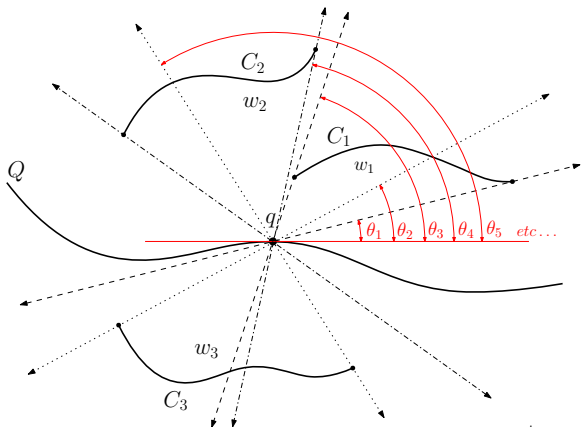


Figure 3: A configuration similar to that shown in Figure 1 for three curves C_1, C_2, and C_3 is depicted, with their respective wedge boundaries extended through the origin. The circular partition induced is shown by the sequence of angles towards the right-hand side of the figure.

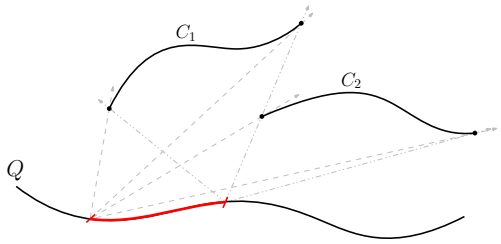


Figure 4: A configuration similar to that shown in Figure 1 for two curves C_1 and C_2 is depicted, the highlighted segment being cyclically invariant with respect to the given population, as can be seen by inspecting the wedge boundaries

Definition 8 (Cyclically Invariant Segments) *A segment along a curve that maintains the same cyclic ordering of boundaries within the circular partitions of each point along its length, is called* cyclically invariant. *Specifically, for a given curve* Q*, a segment* $I \subseteq Q$ *is* cyclically invariant *provided* $\sigma(\mathcal{W}_\mathcal{C})$ *has the same ordering of wedge boundaries as* $\sigma(\mathcal{W}'_\mathcal{C})$*, for all* $\mathcal{W}_\mathcal{C}$ *and* $\mathcal{W}'_\mathcal{C}$ *defined relative to any* $q, q' \in I$ *respectively.*

See Figure 4. Clearly such segments exist when $\{Q\} \cup \mathcal{P}$ is a set of polylines in $\mathbb{R}^2$. This property does not hold more generally for all plane curves[1]. For the remainder of this article, we assume $\{Q\} \cup \mathcal{P}$ is a set of polylines.

[1]We use $\mathcal{C}$ to denote a general set of plane curves, and $\mathcal{P}$ to denote a set of polylines in $\mathbb{R}^2$.

Lemma 1 (Invariant Segments along Polylines) *Given a polyline Q and a set $\mathcal{P}$ of polylines, Q can be partitioned into line segments, each of which is cyclically invariant with respect to $\mathcal{P}$.*

Proof. Consider any line segment L in Q, and assume without loss of generality that every element of $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ lies above the line determined by L. An analogous argument can be applied to polylines that lie below L (any polyline that crosses L can be partitioned into separate polylines above and below L). The tangent points in a circular partition can only undergo a change in relative positions when the reference point (root) q becomes collinear with one of the common tangents between a pair of polylines defining the associated wedges, common to the convex hulls of each polyline. Consequently, as at most four such tangents exist for each pair of polylines, the set of points along L that trigger change in wedge orderings must be finite. Therefore, L can be partitioned into cyclically invariant segments, each of which is a maximal line segment on Q between two consecutive points that trigger changes. □

By Observation 1 and Lemma 1, the double integral in (D.c) can be reformulated as a sum of integrals measuring the total angular area swept out by the wedges of $\mathcal{P}$ with stabbing number weights along all cyclically invariant segments. This reformulation, which is made explicit in Section 3.2, is possible due to the fact that stabbing numbers remain constant within circular partitions, and that the cyclic ordering of each circular partition remains unchanged along each invariant segment.

3 Computing Curve Stabbing Depth for Polylines

In the following section we develop an algorithm for computing the curve stabbing depth of a given polyline Q relative to a given set $\mathcal{P}$ of polylines, based on the identification of critical curve features, such as tangent update points for curve wedges and the partitioning Q into invariant segments.

3.1 One Invariant Segment and One Polyline

We first describe an algorithm for computing the curve stabbing depth of one cyclically invariant segment $I = \overline{q_1 q_2}$ on a query polyline Q relative to another polyline $P = (p_1, p_2, \ldots, p_m)$, before generalizing the algorithm to the complete polyline Q and a set $\mathcal{P}$ of polylines.

The wedge $w(q, P)$ associated with polyline P and a given point q is determined by the tangent points of P (see Figure 5) which can be found by computing the convex hull of P and examining its vertices relative to q using binary search in $O(\log m)$ time. Thus, start by computing the convex hull $\mathrm{CH}(P)$, which can be completed in $O(m \log m)$ total time [14, 6].

In Case 1, begin by deriving the initial tangent points τ_1 and τ_2 of $w(q, P)$ for $q = q_1 \in I$ by using $\mathrm{CH}(P)$ as described in the previous paragraph. Additionally, determine all points of intersection between I and the set of lines corresponding to the extension of all line segments that form, $\partial\,\mathrm{CH}(P)$, the boundary of $\mathrm{CH}(P)$. Denote this set of intersection points along I by T. The points of T signal when and how tangent points of $w(q, P)$ need to be updated as q traverses along I; see Figure 7. The cyclical invariance of I allows the angular area swept out by w along each subsegment $I_i = \overline{ab}$ of I formed by points of T to be evaluated as

$$A_i = \int_{q \in I_i} \theta(w(q, P))\, ds. \tag{WA}$$

We can apply a coordinate transform to render I collinear with the x-axis, which for Case 1(a) using (A.1) in the integral results in (WA) becoming

$$A_i = \int_{a'}^{b'} \left[\tan^{-1}\left(\frac{x - \tau_2(w')_x}{\tau_2(w')_y} \right) - \tan^{-1}\left(\frac{x - \tau_1(w')_x}{\tau_1(w')_y} \right) \right] dx,$$

for the transformed points a', b' and resulting wedge w' defined by the tangent points associated with the points of T delineating I_i. This being an integral with known antiderivative

$$\begin{aligned} A_i = \Bigg[& (\tau_1(w')_x - x) \tan^{-1}\left(\frac{\tau_2(w')_x - x}{\tau_2(w')_y} \right) \\ & + (x - \tau_1(w')_x) \tan^{-1}(\tau_1(w')_y (\tau_1(w')_x - x)) \\ & + \frac{1}{2\tau_1(w')_y} \ln(\tau_1(w')_y (\tau_1(w')_x^2 - 2\tau_1(w')_x x + x^2) + 1) \\ & - \frac{1}{2} \tau_2(w')_y \ln(\tau_2(w')_x^2 - 2\tau_2(w')_x x + \tau_2(w')_y^2 + x^2) \Bigg]_{a'}^{b'}. \end{aligned}$$

As a consequence of the circular partition induced by w being straightforward and w having stabbing number one, we find $D(I, P) = \sum_{I_i \in I} A_i / \pi L(I)$. Analogous analysis can be applied using (A.2) for problems in Case 1(b) reassembling that depicted in Figure 2b.

In Case 2, where $q \in I$ is in the interior of $\mathrm{CH}(P)$, begin by processing P to identify points of self intersection, some of which form closed loops (closed regions). Let L denote the set of points of self intersections of P. The planar subdivision formed by P and $\partial\,\mathrm{CH}(P)$ consists of polygonal faces, each of which can include at most one *window* edge on its boundary, i.e., an edge of $\partial\,\mathrm{CH}(P)$ that is not on P, as well as subpaths of P that do not cross into other faces.[2] This planar subdivision

[2] Observe that the faces of the planar decomposition are effectively simple polygons. Any polyline that protrudes into the interior of a face could be twinned to form a proper simple polygonal face.

can be computed in $O(m^2)$ time using a line segment intersection algorithm (e.g., [2]) and updating a doubly-connected edge list each time a point of intersection is identified.

Next, we construct the shortest geodesic path query data structure given in [15] augmented using the result from [7] in linear time for each face of the planar subdivision, taking $O(m^2)$ total time.

For any endpoint of I within $\mathrm{CH}(P)$ and for every intersection point a between I and the boundary of a face of the planar subdivision (or when I crosses an edge of P while remaining in the same face), we query the two shortest geodesic paths between a and the endpoints of the window edge on $\partial\,\mathrm{CH}(P)$ belonging to the current face. When q is in a face with no window edge, no visibility computation is required as all rays rooted at q stab P. The intersections between I and the extended segments along the shortest paths identify when and which tangent points of the visibility wedges that look out of $\mathrm{CH}(P)$ need to be updated. If the two shortest paths intersect at a vertex of P, then q loses external visibility after one of the two update points corresponding to these extended intersecting segments. Shortest geodesic path queries can be performed in $O(\log m^2 + t)$ time, where t is the number of turns on the reported shortest path. Intersection testing between extended segments and I takes at most $O(t)$ time per path. Thus, this step takes $O(m)$ worst-case time for each such query along I.

The depth for the portion of I within $\mathrm{CH}(P)$ can be calculated as a discrete sum of the depth accumulated by each subsegment I_i of I that result from partitioning I by shortest path update points, by calculating the total wedge area of the difference between 2π and the window visibility wedge at each point along I_i. A calculation that is otherwise analogous to those discussed for Case 1 above.

3.2 A Polyline Q and a Set $\mathcal{P}$ of Polylines

We generalize the algorithm described in Section 3.1 to a query polyline $Q = (q_1, q_2, \ldots, q_m)$ and a set of polylines $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, with $P_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,m})$ for $i = 1, \ldots, n$.

The algorithm is organized into three stages: an initial preprocessing stage applied to $\mathcal{P}$, a separate preprocessing method applied to Q based on results of the first stage, and the final computation of $D(Q, \mathcal{P})$.

Preprocessing $\mathcal{P}$. Begin by computing the convex hull $\mathrm{CH}(P_i)$ of each polyline $P_i \in \mathcal{P}$ to determine wedge tangent points, as done in Section 3.1; see Figure 5. Let $\mathcal{H}$ denote the resulting set of convex hulls. This stage can be completed in $O(nm \log m)$ total time.

Having determined $\mathcal{H}$, compute the collection $\tau(\mathcal{H})$ of all common tangent lines that separate each pair of convex hulls. See Figure 6. That is, compute

$$\tau(\mathcal{H}) = \{\text{lines } l \mid \text{for some } \{P_i, P_j\} \subseteq \mathcal{P} \\ (l \cap \mathrm{CH}(P_i)) \cup (l \cap \mathrm{CH}(P_j)) = \{p_{i,i'}, p_{j,j'}\}\},$$

where $p_{i,i'}$ and $p_{j,j'}$ are vertices of $\mathrm{CH}(P_i)$ and $\mathrm{CH}(P_j)$, respectively.

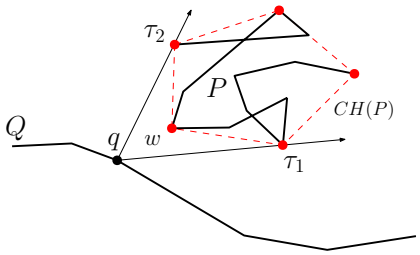


Figure 5: A query polyline Q and tangent points of P highlighted along the boundary of $\mathrm{CH}(P)$. The tangent points and boundary rays for the wedge $w(q, P)$ are also shown.

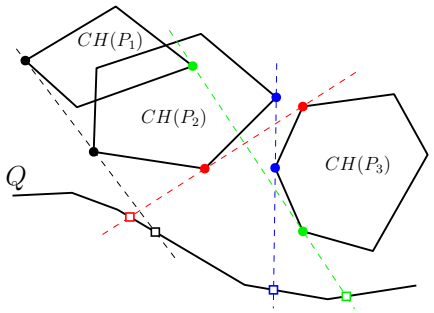


Figure 6: Illustration of the common tangents between convex hulls $\mathrm{CH}(P_1), \mathrm{CH}(P_2)$, and $\mathrm{CH}(P_3)$. To simplify the figure, only those tangents that intersect Q are shown, with their points of intersection marked along Q by boxes.

There are three distinct cases to consider when computing these common tangents: (1) the two convex hulls are disjoint, (2) their boundaries intersect, and (3) one convex hull entirely contains the other. Case 1 is the simplest, in which the common tangents between two convex hulls $\mathrm{CH}(P_1)$ and $\mathrm{CH}(P_2)$ can be computed in $O(\log |\,\mathrm{CH}(P_1)| + \log |\,\mathrm{CH}(P_2)|)$ time [17]. Case 2 requires $O(m)$ time to compute in the worst case. However, if the two convex hull boundaries intersect at most twice, the common tangents can be found in $O(\log(|\,\mathrm{CH}(P_1)| + |\,\mathrm{CH}(P_2)|) \log k)$ time, where $k = \min\{|\,\mathrm{CH}(P_1) \cap \mathrm{CH}(P_2)|, |\,\mathrm{CH}(P_1) \cup \mathrm{CH}(P_2)|\}$ [17]. In Case 3, no computation is performed after identifying that the hulls are nested. It takes up to $O(m)$ time to identify which of the three cases must be applied. Thus, this stage can be computed in $O(n^2 m \log^2 m)$ time.

Preprocessing Q. After preprocessing $\mathcal{P}$, mark the points of intersection between elements of $\tau(\mathcal{H})$ and Q, which, per the proof of Lemma 1, partition Q into cyclically invariant segments. Additionally, as outlined in

Section 3.1, determine all points of intersection between Q and the set of lines corresponding to the extension of all line segments on $\partial\,\mathrm{CH}(P_i)$ for each $i = 1, \ldots, n$. See Figure 7 for the latter. If the intersection between one of these extended segments and Q occurs on the boundary of the convex hull, Q must pass into the interior of the convex hull. Here we enter Case 2 of the algorithm described in Section 3.1, and perform the same computations. In the worst case, Q passes through the convex hulls of all n polylines in $\mathcal{P}$, leading to $O(nm^2)$ worst-case processing time.

This yields two point sets on Q, say S and T, that respectively identify when wedge stabbing numbers and tangent points need to be updated relative to the position of q along Q. Let $\mathcal{I}$ denote the resulting partition of Q into cyclically invariant segments by points of S, after further refinement from the vertices of Q itself. Likewise, for all $I \in \mathcal{I}$, let $I_i \in I$ denote a subdivision of I delineated by tangent update points of T. There are at most $O(n^2)$ many points in S as there are at most four common tangents for each pair of convex hulls. Additionally, there are at most $O(m)$ segments composing each of the n convex hulls, and $O(m^2)$ internal update points for each crossed convex hull, so T contains at most $O(nm^2)$ points. Consequently, this step takes $O(n^2m + nm^2)$ worst-case time to compute all possible intersections.

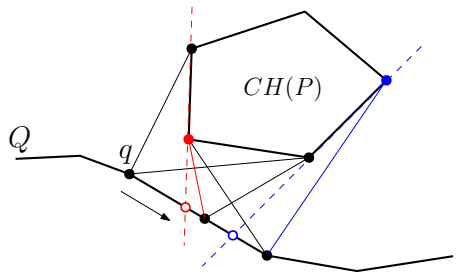


Figure 7: Depiction of a query polyline Q with tangent update points of a polyline P shown along its length as open circles. These points are derived from the intersection between Q and lines passing though the parameter segments of $\mathrm{CH}(P)$. Only those lines that intersect Q are shown. Then q traverses the length of Q (in the indicated direction) the tangent points of the wedge $w(q, P)$ change whenever one such point is crossed. The update points are color matched with the resulting tangent point (line) change.

Computing the curve stabbing depth of Q. Let $\vec{u}$ denote the unit direction vector associated with a line segment $I = \overline{q_1q_2} \in \mathcal{I}$ along Q. Construct the matrix $P_{\vec{u}} + B$, composed of the transition matrix $P_{\vec{u}}$ from the standard basis of $\mathbb{R}^2$ to the orthonormal basis $\{\vec{u}, -1/\vec{u}\}$, and a vertical translation matrix B that displaces I to have height zero after the transformation. Applying this transformation to $\mathcal{P}$ pointwise for each $I \in \mathcal{I}$ allows us to calculate the area swept out by wedges along a path as described in Section 3.1. Let $\mathcal{P}' = \{P'_1, P'_2, \ldots, P'_n\}$ be the set of transformed polylines.

Starting at q_1, construct the set of wedges $\mathcal{W}_{\mathcal{P}'}$. This is accomplished by calculating the tangent points of each convex hull within $\mathcal{H}$ relative to q_1 using binary search in $O(n \log m)$ time. The set $\mathcal{W}_{\mathcal{P}'}$ is updated incrementally by monitoring the points of T crossed by q while traversing I. Each update takes $O(1)$ time by walking one vertex clockwise or counterclockwise around the perimeter of the convex hull depending on the relative motion between q translating along I and the convex hull.

Afterwards, construct $\sigma(\mathcal{W}_{\mathcal{P}'}) = (\theta_0, \theta_1, \ldots, \theta_{4n})$ by sorting the lines associated to each tangent point by slope, treating the portion of the line extended through the origin separately. During this process, take note of which regions overlap to calculate the stabbing numbers of each angular region in the partition (subdivided wedges) as in (1) and Observation 1. These stabbing numbers are iteratively updated by monitoring the points of S crossed in $O(1)$ time per event as is done for tangent points above. From the circular partition $\sigma(\mathcal{W}_{\mathcal{P}'})$, select a minimizing subset iteratively by defining the indicator variable (bit sequence)

$$\delta_i = \begin{cases} 1 & \text{if } \mathrm{stab}_{\mathcal{P}}(\overrightarrow{q_{\theta^*}}) \leq \mathrm{stab}_{\mathcal{P}}(\overrightarrow{q_{\theta^*+\pi}}) \\ & \quad \text{for } \theta_{i-1} \leq \theta^* \leq \theta_i \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, 4n$. This selection procedure performs the same task as the minimization operation within (D.c).

At last, we can compute the depth of Q accumulated along I, by reformulating (D.c) in terms of summations over all $I_i \in I$, specifically,

$$D_I = \frac{1}{\pi L(Q)} \sum_{I_i \in \mathcal{I}} \sum_{j=1}^{4n} \delta_j \,\mathrm{stab}_{\mathcal{P}'}(\overrightarrow{q_{\theta_j^*}}) A_j, \qquad \text{(D.i)}$$

for any q along I_i and sample angle $\theta_j^* \in [\theta_{j-1}, \theta_j)$, and the angular area A_j swept out by the wedge bounded between the angles $[\theta_{j-1}, \theta_j]$ while q is translated across I_i, as calculated above using (WA).

The total depth of Q is found by evaluating the sum $D(Q, \mathcal{P}) = \sum_{I \in \mathcal{I}} D_I$.

Forming the partition $\sigma(\mathcal{W}_{\mathcal{P}'})$ and selecting the chosen subset takes at most $O(n \log m)$ time. The query polyline Q contains at most $O(n^2 + nm^2)$ update points from S and T which are used during the computation of D_I, and at most $O(m)$ directional transitions from its m constitutional line segments where each transformation to the set $\mathcal{P}$ takes $O(nm^2)$ time. Thus, this final stage takes $O(n^2 + nm^2)$ time.

These results are summarized in the following theorem:

Theorem 2 *Given an m-vertex polyline Q and a set $\mathcal{P}$ of n polylines, each with m vertices, we can compute the curve stabbing depth of Q relative to $\mathcal{P}$ in $O(nm^2 + n^2 m \log^2 m)$ time.*

Expressed differently, the running time is $O(k^2)$, where k denotes the total number of vertices in the input polylines $\{Q\} \cup \mathcal{P}$.

4 Discussion and Directions for Future Research

In this section, we discuss depth medians, possible generalizations of curve stabbing depth to higher dimensions, and other possible measures of curve depth. Due to space constraints, discussion of properties has been omitted (e.g., stability, robustness, etc.).

4.1 Median Curves and Depth Median Points

The depth for any particular curve in a set can be computed by treating it as a query curve Q. The comparison of all the resulting depth scores allows for a median outwards ranking of all curves.

Moreover, observe not all points along the length of a curve Q contribute equally to the curve stabbing depth of Q relative to the set $\mathcal{C}$ of curves. The depth of a point (a degenerate curve) is given by (D.p). It follows that for some point q on Q, $D(q,\mathcal{C}) \geq D(Q,\mathcal{C})$. Consequently, this gives:

Observation 2 *For any given set $\mathcal{C}$ of plane curves, there exists a point $m \in \mathbb{R}^2$ that is a depth median of $\mathcal{C}$. That is,*

$$D(m,\mathcal{C}) = \max_{Q \in \mathcal{Q}} D(Q,\mathcal{C}),$$

where $\mathcal{Q}$ denotes the set of all plane curves.

4.2 Generalizations to Higher Dimensions

When a curve Q and a set $\mathcal{C}$ of curves lie in a k-dimensional flat of $\mathbb{R}^d$ for some $k < d$, the d-dimensional curve stabbing depth of Q as calculated using a ray relative to $\mathcal{C}$ is zero; whereas, the k-dimensional curve stabbing depth of Q relative to $\mathcal{C}$ is non-zero in general, meaning that the straightforward generalization of Definition 4 is not consistent across dimensions.

Alternatively, another natural generalization of Definition 4 to higher dimensions is to replace the rotating stabbing ray by a k-dimensional half-hyperplane, and to measure the number of curves it intersects as it rotates. This second generalization is consistent across dimensions.

4.3 Alternative Definitions

Alternative possible definitions for the stabbing depth of curves considered by the authors include:

$$\int_{q \in Q} \min_{0 \leq \theta < \pi} \min\{\text{stab}_{\mathcal{C}}(\overrightarrow{q_\theta}), \text{stab}_{\mathcal{C}}(\overrightarrow{q_{\theta+\pi}})\}\, ds, \qquad (2)$$

which differs from (D.c) by a minimum in place of the second integral (maximum was also considered). Equation (2) often gives a zero depth value regardless of the position of Q relative to $\mathcal{C}$. For example, consider a set $\mathcal{C}'$ of n parallel line segments of equal length. Each of these line segments has depth zero relative to $\mathcal{C}'$ by (2) because every point on every segment is the root of some ray that does not intersect any other segment in $\mathcal{C}'$. Conversely, using Definition 4 instead, the line segment at the centre (median) of $\mathcal{C}'$ has greatest depth, with depth values decreasing monotonically toward the two line segments on the outside of $\mathcal{C}'$, which are the only two curves in $\mathcal{C}'$ with depth zero.

4.4 Approximation Algorithms using Randomization

Definition 4 suggests that efficient approximate computation by Monte Carlo methods is likely possible using a random sample of rays rooted along the query curve Q. One possible direction for future research is to bound the expected quality of approximation and the expected running time as functions of the number of random rays selected.

4.5 Upper Envelopes of Sets of Pseudolines

Our algorithm for computing curve stabbing depth involves identifying the extreme points of each curve $P \in \mathcal{P}$ relative to a point q that follows the query curve Q. When P is a polyline, the extreme points can be identified by computing the upper and lower envelopes of the angle formed by each vertex of P relative to q as a function of the position of q on Q. These functions are a set of pseudolines when Q is a line segment; it may be possible to compute upper and lower envelopes of this set efficiently by constructing the convex hull of a set of points dual to the set of pseudolines (e.g., [1]), which may lead to a simpler and more efficient algorithm for computing curve stabbing depth.

References

[1] P. K. Agarwal and M. Sharir. Pseudo-line arrangements: Duality, algorithms, and applications. *SIAM Journal on Computing*, 34(3):526–552, 2005.

[2] I. J. Balaban. An optimal algorithm for finding segments intersections. In *Proceedings of the 11th Symposium on Computational Geometry (SoCG)*, pages 211—219, 1995.

[3] V. Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A (General)*, 139(3):318–355, 1976.

[4] K. Buchin, M. Buchin, M. van Kreveld, M. Löffler, R. I. Silveira, C. Wenk, and L. Wiratma. Median trajectories. *Algorithmica*, 66:595–614, 2013.

[5] K. Buchin, M. Buchin, M. J. van Kreveld, B. Speckmann, and F. Staals. Trajectory grouping structure. *Journal of Computational Geometry*, 6(1):75–98, 2015.

[6] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16:361–368, 1996.

[7] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.

[8] J. Chu, X. Ji, Y. Li, and C. Ruan. Center trajectory extraction algorithm based on multidimensional hierarchical clustering. *Journal of Mechatronics and Artificial Intelligence in Engineering*, 2(2):63–72, 2021.

[9] G. Claeskens, M. Hubert, L. Slaets, and L. Vakili. Multivariate functional halfspace depth. *Journal of the American Statistical Association*, 109(505):411–423, 2014.

[10] F. A. Cuevas and R. Fraiman. Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics*, 22:481–496, 2007.

[11] P. L. de Micheaux, P. Mozharovskyi, and M. Vimond. Depth for curve data and applications. *Journal of the American Statistical Association*, 116(536):1881–1897, 2021.

[12] S. Durocher and M. Y. Hassan. Clustering moving entities in euclidean space. In *Proceedings of the 17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 162 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:14, 2020.

[13] F. Ferraty and P. Vieu. Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1-2):161–173, 2003.

[14] R. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972.

[15] L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989.

[16] F. Ieva and A. M. Paganoni. Depth measures for multivariate functional data. *Communications in Statistics - Theory and Methods*, 42(7):1265–1276, 2013.

[17] D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures (WADS)*, pages 183–193, Berlin, Heidelberg, 1995. Springer-Verlag.

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

[19] R. Y. Liu, J. M. Parelius, and K. Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–840, 1999.

[20] H. Oja. Descriptive statistics for multivariate distributions. *Statistics & Probability Letters*, 1(6):327–332, 1983.

[21] P. J. Rousseeuw and M. Hubert. Regression depth. *Journal of the Amerian Statistical Association*, 94(446):388–402, 1999.

[22] R. Serfling. Depth functions in nonparametric multivariate inference. In R. Y. Liu, R. Serfling, and D. L. Souvaine, editors, *Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications, Proceedings of a DIMACS Workshop*, volume 72 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–16, 2003.

[23] J. W. Tukey. Mathematics and the picturing of data. In R. D. James, editor, *Proceedings of the International Congress of Mathematicians*, volume 2, pages 523–531, 1975.