Inclusion Depth for Contour Ensembles

Nicolas F. Chaves-de-Plaza, Prerak Mody, Marius Staring,
René van Egmond, Anna Vilanova and Klaus Hildebrandt

***Abstract*—Ensembles of contours arise in various applications like simulation, computer-aided design, and semantic segmentation. Uncovering ensemble patterns and analyzing individual members is a challenging task that suffers from clutter. Ensemble statistical summarization can alleviate this issue by permitting analyzing ensembles' distributional components like the mean and median, confidence intervals, and outliers. Contour boxplots, powered by Contour Band Depth (CBD), are a popular non-parametric ensemble summarization method that benefits from CBD's generality, robustness, and theoretical properties. In this work, we introduce Inclusion Depth (ID), a new notion of contour depth with three defining characteristics. First, ID is a generalization of functional Half-Region Depth, which offers several theoretical guarantees. Second, ID relies on a simple principle: the inside/outside relationships between contours. This facilitates implementing ID and understanding its results. Third, the computational complexity of ID scales quadratically in the number of members of the ensemble, improving CBD's cubic complexity. This also in practice speeds up the computation enabling the use of ID for exploring large contour ensembles or in contexts requiring multiple depth evaluations like clustering. In a series of experiments on synthetic data and case studies with meteorological and segmentation data, we evaluate ID's performance and demonstrate its capabilities for the visual analysis of contour ensembles.**

***Index Terms*—Uncertainty visualization, contours, ensemble summarization, depth statistics.**

I. Introduction

Different applications in simulation, computer-aided design, and semantic segmentation have to deal with ensembles of curves. Analyzing these ensembles permits understanding uncertainties in the results. We focus on ensembles of spatiotemporal scalar fields from which one can extract contours, closed and consistently-oriented curves. These appear in several domains. One example is meteorology, where analysts use ensembles of weather forecasts to analyze the predictions' variability under different initial conditions or changes in the computational model [1]. Another example is semantic segmentation, where ensembles are used to quantify the uncertainty that might come from the training data or the model [2]. In image-guided medical specialties, ensembles of segmentations are analyzed for planning the patients' treatments [3].

Visual inspection of the ensemble can facilitate its analysis and understanding. Spaghetti plots, which draw each contour in the ensemble using a different color, are a popular technique. They are attractive because they are accessible, represent all the data, and are simple to implement. Nevertheless, as the ensemble size increases, spaghetti plots become cluttered, potentially hiding interesting features of the ensemble. Motivated by these limitations, ensemble summarization methods have been proposed. They reduce information by extracting features of interest, such as representative members and contour variability, from the ensemble and visualize them using visual encodings based on lines and bands [4]–[7].

A successful contour summarization technique is the contour boxplot (CBP) [7], which has been used in the fields of meteorology [8] and medicine [9]–[11]. As Fig. 1 illustrates, like traditional boxplots, CBPs depict four statistical features of an ensemble: the median, the trimmed mean, confidence intervals, and outliers. Underlying the CBP is the concept of statistical depth, which extends univariate order and rank statistics to complex multivariate datasets by establishing a center-outward measure of centrality for the ensemble members [12].

Contour Band Depth (CBD) was jointly presented with the CBP visual idiom [7]. CBD draws inspiration from previous developments in statistical functional depth, generalizing Band Depth [13] to the contour case. Computing a contour's CBD entails finding the number of times it falls within the bands formed by all J-sized subsets of ensemble contours. The depth is maximum when the contour falls within all bands and decreases as fewer bands contain it. CBD is easy to implement and provides several theoretical guarantees. Unfortunately, CBD's requirement of comparing a contour against bands formed by all other contours makes it computationally expensive. When considering bands made of pairs of contours ($J = 2$), as recommended in [7], CBD takes $\mathcal{O}(N^3)$ operations to compute the depths of an N-member ensemble, rendering the method impractical for larger ensembles.

In this paper, we propose an alternative notion of contour depth called Inclusion Depth (ID). ID contributes to the arsenal of depth-based contour analysis methods in three ways.

First, ID provides a novel statistical depth for ensembles of contours. It draws inspiration from Half-Region Depth (HRD) and generalizes HRD from the class of functions to contours. This connection to HRD endows ID with theoretical properties and enables computational advantages analogous to those of HRD, also for ensembles of contours. HRD operates on graphs of functions with a natural parametrization and one-to-one correspondence given by the function's domain, for example, the unit interval in $\mathbb{R}$. Extending this approach to the case of contours is not trivial because of the lack of an independent variable that establishes correspondences between the contours and because contours require topological considerations, like how to handle disconnected components. In Section IV, we

Nicolas F. Chaves-de-Plaza is the corresponding author. e-mail: n.f.chavesdeplaza@tudelft.nl

Nicolas F. Chaves-de-Plaza, René van Egmond, and Klaus Hildebrandt are with TU Delft, Netherlands.

Prerak Mody and Marius Staring are with Leiden University Medical Center, Netherlands.

Anna Vilanova is with TU Eindhoven, Netherlands.

Nicolas F. Chaves-de-Plaza and Prerak Mody are with HollandPTC, Netherlands.

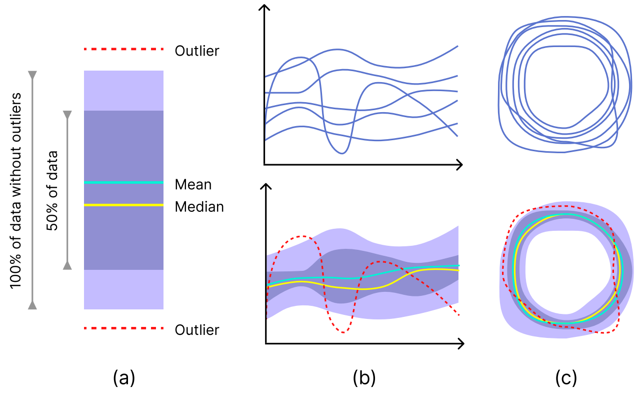


Fig. 1. Extension of the boxplot idiom (a) to the functional (b) and contour (c) data types.

present the ID framework, detailing how it overcomes the challenges that the case of contours brings.

Second, ID leverages a simple principle that makes it accessible and facilitates the interpretation of the results. Specifically, ID leverages the inside/outside relationships between contours to estimate the ensemble's depth. To compute a contour's ID we compute how many other contours of the ensemble the contour contains and in how many other contours it is contained. Intuitively, a highly central contour has similar values for both quantities. An outlier might have an asymmetry of these quantities, if it's a magnitude outlier, or lower values for both, in the case of a shape outlier.

Third, the computation of ID scales better than the CBD with respect to the number of members in the ensemble. ID requires $\mathcal{O}(N^2)$ operations while CBD needs at least $\mathcal{O}(N^3)$. The reason is that ID considers all pairs of contours, while CBD processes all triplets or even quadruples. In Section VI, we evaluate ID, empirically showing that performing only pairwise comparisons does not degrade ID's performance and yields depth scores qualitatively comparable to CBD's.

We further demonstrate the practical use of ID in Section VII by performing depth-based exploratory analysis of several real datasets from diverse domains like segmentation in radiotherapy and meteorological forecasting. Based on the results, we expect the faster but still performant ID will enable visual analysis of larger ensembles using depth-based visualizations like CBP, which allows both quantitative and qualitative interpretation of contour ensembles. Furthermore, it will bring applications that require multiple or/and fast depth evaluations like regression [14] and clustering [15] within reach.

II. Related Work

Our method fits in the context of uncertainty visualization. Ensembles permit quantifying predictive uncertainties due to changes in the initial conditions, the training data, or the model parameters [2]. Wang et al. [16] survey ensemble visualization techniques based on the data type, visualization method, and analytic task.

There are several alternatives to present a visual overview of contour ensembles. Spaghetti plots are a composition-after-visualization technique that plots each contour using a different color [17]. Although straightforward to implement and interpret, spaghetti plots become cluttered as the size of the ensemble grows, potentially hiding trends and interesting members. To address this issue, several ensemble summarization techniques have been proposed in recent years that aggregate contour data into salient features before visualizing it. Most available summarization techniques share a visual language that uses contour lines for the ensembles' representative members like the median, mean, and outliers, and bands for areas of interest like the ensemble's spread [18] and confidence intervals [7].

Available summarization techniques differ in the features they compute and the assumptions they make. Parametric model-based techniques assume a data distribution and use available models to derive statistical quantities. Ferstl et al. [5], [19] fit a Gaussian distribution on the contours' PCA-reduced signed distance field (SDF) transform and use it to derive a median and calculate bands. In [20], Pothkow and Hege use a Gaussian model to describe each grid point and use this model together with iso-contour density and level-cross probability to extract the iso-contours probability density. Parametric techniques are conceptually attractive as they permit extracting information analytically [5]. Nevertheless, they impose assumptions on the data, like normality, which limits the applicability in practice. Our method is fully non-parametric using a depth-based ordering of the contours to detect outliers and derive quantities of interest the median and the trimmed mean.

The family of data-based non-parametric methods does not impose assumptions on the data distribution and, therefore, can describe the ensemble data on each point more accurately [21], [22]. Local summarization methods operate on the grid in which contours lie, computing point-wise statistics. Examples are contour probability plots, which extract bands by thresholding a scalar field of percentages [18], and EnConVis [4], which performs point-wise kernel density estimation, and then uses the per-point density to extract bands and representatives. Contour grid points are not independent of each other, so computing summaries based solely on point-wise estimates can fail to consider global characteristics of the contour data like the topological relationships between contours.

Demir et al. [6] use a vector-to-closest-point representation along the contours boundary points to quantify their centrality based on the vector lengths and directions. Their approach requires only comparisons between contours, making it more efficient than CBD. Nevertheless, it uses parametric statistical models that require parameter fitting to obtain the centrality estimates. Furthermore, it is unclear how the method performs under different ensemble distributions, which makes it hard to compare to existing contour depth methods like CBD.

III. Background: Contour Depth and Boxplots

A. Statistical Depth

Statistical depth provides a framework for extending concepts like the median, trimmed mean, and outliers, which depend on the points' ranks and orderings from the univariate to the multivariate case. Given a cloud of N d-dimensional points $X \in \mathbb{R}^{N \times d}$, a depth function $D(z, X) : \mathbb{R}^d \rightarrow [0, 1]$ yields a center-outward measure of the centrality or depth of a point z with respect to X. Intuitively, the farther away a point z is from the center of X, the lower its centrality. In practice, there are different methods for computing $D(z, X)$, which come with different guarantees in terms of the function's behavior like invariance to different geometric transformations of X [12].

Statistical depth functions were originally devised to handle multivariate data. Nevertheless, their performance decreases with a high number of dimensions. Furthermore, in some cases, data is more naturally represented as functions. In response to these observations, several definitions of depth that apply to functional data have been recently proposed [13], [23]. Two predominant functional depth methods are Band Depth (BD) [13] and Half-Region Depth (HRD) [23]. Inspired by the multivariate simplicial depth [24], BD computes a function's depth by comparing it to the bands formed by all other subsets of functions in the ensemble. Contour Band Depth, presented in the next subsection, generalizes BD's formulation and extends it to the case of contours.

Instead of forming bands, HRD looks at the proportion of functions lying on each side of the function of interest to determine its depth. The multivariate analog of HRD is Tukey's half-space depth [25]. HDR is more computationally efficient than BD, requiring only $\mathcal{O}(N)$ comparisons per function. Furthermore, it has been shown to yield comparable depths to BD [23]. The proposed Inclusion Depth generalizes HDR's formulation and extends it to the case of contours. In the following, we outline HDR.

Let $X = \{x_1, x_2, ..., x_N\}$ with $x_i : I \rightarrow \mathbb{R}$ be an ensemble of functions defined on the compact interval I. The graph of a function $x \in X$ can be defined as

$$G(x) = \{(t, x(t)), t \in I\} \tag{1}$$

The epi and hypographs of x, which correspond to the regions above and below $G(x)$, can be defined as

$$\begin{aligned} hyp(x) &= \{(t, y) \in I \times \mathbb{R} : y \leq x(t)\}, \\ epi(x) &= \{(t, y) \in I \times \mathbb{R} : y \geq x(t)\}. \end{aligned} \tag{2}$$

The HRD of x can be computed by evaluating the proportion of times $G(x)$ is contained in the epi and hypographs of other functions of the ensemble. Formally,

$$\text{HRD}(x|X) = \min\{\text{IN}_{hyp}(x), \text{IN}_{epi}(x)\}, \tag{3}$$

where

$$\begin{aligned} \text{IN}_{hyp}(x) &= \frac{1}{N} \sum_{i=1}^{N} G(x) \subset hyp(x_i), \\ \text{IN}_{epi}(x) &= \frac{1}{N} \sum_{i=1}^{N} G(x) \subset epi(x_i), \end{aligned} \tag{4}$$

where $A \subset B$ is 1 if A is contained in B and 0 otherwise.

HRD in Eq. 3 attains its maximum value of 0.5 when $G(x)$ is contained in as many epi and hypographs of the other functions in the ensemble. The HDR satisfies several of the properties of a valid depth function [26]: linear invariance, maximality at the center, monotonically decreasing on rays, and upper-semicontinuity. Finally, a finite-dimensional version can be obtained by drawing d samples from I. When $d = 1$, the Half-Region Depth is equivalent to the Tukey depth.

B. Contour Band Depth

Statistical depth allows for robust and model-free exploratory data analysis. Contour Band Depth (CBD) permits applying the depth methodology to contours [7]. Similarly to functional BD, CBD computes a contour's depth by determining how many bands formed by all other possible J-sized contour subsets (where $J \in \mathbb{Z}$ and $J \geq 2$) contain the contour. A contour is in a band if it contains the intersection of the band's contours and is contained in their union. To reduce the computational cost of verifying contour containment in $\sum_{i=2}^{N} \binom{N}{i}$ bands (where N is the size of the contour ensemble), the authors propose to use $J = 2$. To alleviate the tendency of CBD with $J = 2$ to produce depth ties, the authors propose a modified CBD (mCBD). Instead of strictly enforcing the containment property, mCBD considers the proportion of the contour that falls outside each band when computing its depth. CBD and mCBD compute an ensemble's depths in $\mathcal{O}(N^3)$ time

C. Contour Boxplots

Boxplots offer a visualization of a dataset's summary statistics. Specifically, as Fig. 1 illustrates, a boxplot has four components. The gold and blue-colored lines represent the median and the trimmed mean, respectively. The trimmed mean is the average of the dataset with the outliers removed. Purple bands around the mean encode the interquartile range. Finally, outliers are shown using red dashed lines. As the middle and right side of Fig. 1 shows, the idea of boxplots can be extended to ensembles of functional [27] and contour [7] types through the concept of functional and contour depth. In these cases, the per-member depth values are used to compute the different statistics. The median is the member with the highest depth value and the interquartile ranges are bands formed by members whose depths fall in the specified ranges. Finally, the members with the lowest depths are flagged as outliers.

IV. Inclusion Depth

In this section, we introduce Inclusion Depth (ID). While ID can be defined for contours in $\mathbb{R}^2$ and $\mathbb{R}^3$, for sake of simplicity, we consider the two-dimensional case.

Let $C = \{c_1, c_2, ..., c_N\}$ be an ensemble of contours, where a contour c_i is a pair of a function $F_i : \Omega \rightarrow \mathbb{R}$ and an isovalue $q_i \in \mathbb{R}$. Here Ω is a compact domain in $\mathbb{R}^2$, such as rectangle. A contour encloses a subset in the plane that we call the inside region

$$in(c_i) = \{p \in \Omega | F_i(p) < q_i\}. \tag{5}$$

ID is based on a simple principle. We evaluate for all pairs $c_i, c_j \in C$ whether or not $in(c_i)$ is contained in $in(c_j)$. Then,

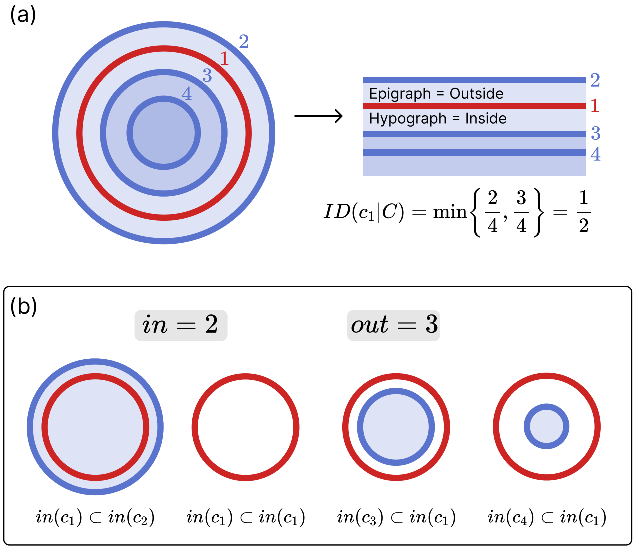

(a)
2
1
3
4
Epigraph = Outside
Hypograph = Inside
2
1
3
4
$ID(c_1|C) = \min\left\{\frac{2}{4}, \frac{3}{4}\right\} = \frac{1}{2}$
(b)
$in = 2$
$out = 3$
$in(c_1) \subset in(c_2)$
$in(c_1) \subset in(c_1)$
$in(c_3) \subset in(c_1)$
$in(c_4) \subset in(c_1)$

Fig. 2. Example of the ID computation for a 4-contour ensemble. In red is the contour for which we are currently estimating the depth. (b) shows the four comparisons that need to be performed to compute ID based on Eq. 8. Note that $c1 \subset c1$ (second column) counts for the inside and outside relationships.

we form the fraction of contours of C in which $in(c_i)$ is contained,

$$\text{IN}_{in}(c_i) = \frac{1}{N}\sum_{j=1}^{N} in(c_i) \subset in(c_j), \qquad (6)$$

and the fraction of contours of C that are contained $in(c_i)$,

$$\text{IN}_{out}(c_i) = \frac{1}{N}\sum_{j=1}^{N} in(c_j) \subset in(c_i). \qquad (7)$$

In the sums in Eqs. 6 and 7, we interpret $in(c_i) \subset in(c_j)$ as the numerical value 1 if $in(c_i)$ is contained in $in(c_j)$ and as 0 otherwise. The ID is the minimum of the two fractions

$$\text{ID}(c_i|C) = \min\{\text{IN}_{in}(c_i), \text{IN}_{out}(c_i)\}. \qquad (8)$$

Fig. 2 illustrates the process of computing the ID of a contour. As the top panel depicts, ID is related to HRD. Specifically, the proof sketch in the appendix shows that if there is an invertible transform mapping the contours to graphs of functions, our definition of ID is the same as HRD. Furthermore, ID is a generalization of HDR. Applying ID to the region that contains the functions' graphs is equivalent to computing the depth of the functions ensemble using HRD.

ID is more general than HDR, accommodating the different topologies that arise in higher dimensions. Fig. 3 shows examples of how ID deals with different cases. Note that, by subset operations, the definitions of $\text{IN}_{in}(c)$ and $\text{IN}_{out}(c)$ in Eqs. 6 and 7 ensure that the two contours under comparison are nested. As the bottom right panel of Figure 3 shows, when contours are not nested, the comparison will not add to the inside or outside counts, effectively reducing the depth of the contour under consideration. ID has other interesting properties for which we sketch proofs in the appendix. First, ID's results are invariant to homeomorphic transformations of the domain Ω, a general class of transformations that includes affine transformations and Möbius transformations. Second, ID's results are invariant to the choice of inside and outside. Finally, if contours are nested, the contour with median size attains the largest depth and the depth vanishes when the contour' size tends to zero and infinity.

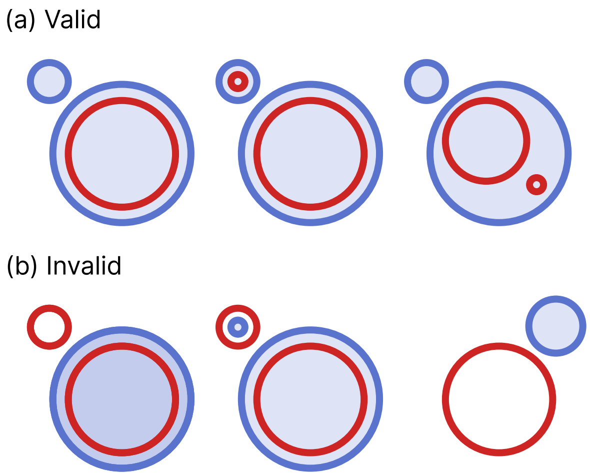


Fig. 3. Examples of how ID deals with different cases. If contours are nested (a), their relationship will add to the inside/outside counts. In other cases (b), the inside/outside counters will not increase, effectively reducing the overall depth.

Algorithm 1 shows how to compute the ID of a contour ensemble. For computations, we assume Ω to be a rectangle, *e.g.*, the bounding box of the ensemble of contours, and discretize the rectangle by a regular grid. ID's scaling behavior depends mainly on the ensemble's size (N). Nevertheless, the grid size will also impact the algorithm's scaling behavior when performing the inside/outside comparisons. Given that CBD shares this cost, we assume the grid size is constant. Under this assumption, ID has a computational complexity of $\mathcal{O}(N^2)$, which is a significant improvement over the $\mathcal{O}(N^3)$ complexity of CBD.

Algorithm 1 Inclusion Depth (ID)

```
Require: $C, N$                          ▷ Contour ensemble, number of contours
  $\mathbf{d}^{\text{ID}} \leftarrow \{\}$                                   ▷ Inclusion depths
  for $i = 1$ to $N$ do
      $num\_in \leftarrow 0; num\_out \leftarrow 0$        ▷ Inside/outside counts
      for $j = 1$ to $N$ do
          $num\_in \leftarrow num\_in + [in(c_i) \subset in(c_j)]$
          $num\_out \leftarrow num\_out + [in(c_j) \subset in(c_i)]$
      end for
      $\text{IN}_{in}(c_i) = num\_in/N$
      $\text{IN}_{out}(c_i) = num\_out/N$
      $\mathbf{d}^{\text{ID}} \leftarrow \mathbf{d}^{\text{ID}} \bigcup \min\{\text{IN}_{in}(c_i), \text{IN}_{out}(c_i)\}$
  end for
  return $\mathbf{d}^{\text{ID}}$
```

V. Epsilon Inclusion Depth

If the ensemble's contours are non-smooth and intersect, inside/outside relationships will be ambiguous. In these cases, ID will produce ties and low-depth scores that reduce the method's practical utility. In this section, we present the Epsilon Inclusion Depth (eID) that relaxes the definitions of inside/outside in ID, reducing the effect of highly varying contours on the depth estimate.

For this extension, we proceed analogously to HRD, for which modified HRD (mHRD) alleviates the problem that strongly varying functions pose on HRD by relaxing the requirement that the graph of a function must lie entirely in the epi or hypograph. mHRD determines the average proportion of the domain that a function's graph lies in the hypo and epigraphs of other functions [23]. This strategy is not directly applicable to the case of contours because of the lack of an independent variable. Therefore, we follow a strategy inspired by the modified Contour Band Depth in [7], which operates directly on the contours' domain and therefore does not require a dependent variable.

First, we define the epsilon subset operator $A \subset_\epsilon B$ for two sets $A, B \subset \mathbb{R}^2$. In contrast to the subset operator $\subset$, which returns either 0 or 1, $\subset_\epsilon$ yields a value in the interval $[0, 1]$. It is defined as

$$A \subset_\epsilon B = 1 - \begin{cases} 0 & |A| = 0, \\ |A - B|/|A| & \text{otherwise,} \end{cases} \tag{9}$$

where $|A|$ denotes the area of A and $A - B$ the set difference. Note that $A \subset_\epsilon B$ will be one if B contains A. If a part of A lies outside of B, $\subset_\epsilon$ will yield lower values.

The definition of eID is analogous to ID except that the $\subset$ operator is replaced by the $\subset_\epsilon$ operator. We consider the values

$$\begin{aligned} \mathrm{IN}^{\epsilon}_{in}(c_i) &= \frac{1}{N}\sum_{j=1}^{N} in(c_i) \subset_\epsilon in(c_j), \\ \mathrm{IN}^{\epsilon}_{out}(c_i) &= \frac{1}{N}\sum_{j=1}^{N} in(c_j) \subset_\epsilon in(c_i). \end{aligned} \tag{10}$$

The eID is the minimum of the two values

$$\epsilon\mathrm{ID}(c|C) = \min\{IN^{\epsilon}_{in}(c), IN^{\epsilon}_{out}(c)\}. \tag{11}$$

Fig. 4 shows how $\subset_\epsilon$ works across a variety of cases. As the extremes of the first row illustrate, when $in(c_i)$ (red) is completely inside or outside of c_j (blue), the difference between $in(c_i) \subset_\epsilon in(c_j)$ and $in(c_j) \subset_\epsilon in(c_i)$ is the largest. When the relationship between the contours is ambiguous, the second row of the figure shows that the difference shrinks. Also, the values of these quantities decrease, which has the effect of reducing the contribution of the c_i/c_j comparison to the overall depth calculation. Finally, eID is invariant to area-preserving transformations. We sketch the proof of this property in the appendix.

As the next sections show, eID provides meaningful results even when contours have many intersections. The implementation of eID only requires swapping $\subset$ for $\subset_\epsilon$ in Alg. 1. eID maintains ID's computational complexity of $\mathcal{O}(N^2)$.

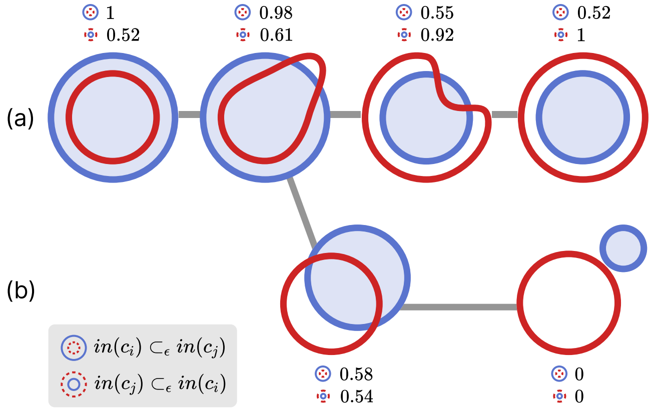

1
0.52
0.98
0.61
0.55
0.92
0.52
1
(a)
(b)
$in(c_i) \subset_\epsilon in(c_j)$
$in(c_j) \subset_\epsilon in(c_i)$
0.58
0.54
0
0

Fig. 4. Examples of computing the inside and outside relationships with the $\subset_\epsilon$ operator in Equation 9 for different contour configurations. In red and blue are contour $\{c_i, c_j \ \in C\}$. The first row shows the transition of c_i from being completely inside to completely outside of c_j. The second row shows the values that $\subset \epsilon$ yields in ambiguous cases.

VI. Experiments

In this section, we perform an extensive evaluation of the Inclusion Depth (ID) method using synthetic data. Specifically, we assess the scaling behavior of ID as the dataset's size increases and investigate the robustness of estimators derived with ID and the method's performance at identifying outliers. Before continuing with the experiments, we detail our experimental setup.

A. Experimental Setup

In the experiments, we compare ID and eID against the other existing contour depth method: Contour Band Depth (CBD). We consider the strict and modified (mCBD) versions of CBD. The only parameter of CBD is the number J of contours used to form the band. We set it to $J = 2$ for all the experiments.

We implemented CBD and ID methods in a common Python-based framework. All methods receive as input a list of binary masks corresponding to a discretization of Eq. 5. These binary masks can be obtained, for example, as the output of a segmentation algorithm or by thresholding scalar fields using an iso-value. We represent these masks as integer Numpy [28] two-dimensional arrays with a resolution of 300×300 pixels and isotropic pixel spacing. We use Numpy's built-in functions to perform operations on the masks, like finding their union and intersection. It is possible to accelerate CBD and ID by, for instance, parallelizing the methods' outer loop. We chose not to implement such optimizations and focus on the methods' asymptotic algorithmic scaling.

Similar to [7], we use synthetic ensembles of circular shapes contaminated with outliers to assess the methods' performance. We extend the experiments of contour depth by considering different types of outliers separately, following the experimental paradigm used to evaluate the functional Half-Region Depth [23]. The first row of Figure 5 showcases the different outliers we consider.

To generate ensembles of contours contaminated with outliers, we define a stochastic model from which we can sample shapes. The model results from a mixture of a base model r_0 and a second model r_1, which depends on the outlier type under consideration. For both r_0 and r_1, we use stochastic processes indexed by the shape's angle, yielding angle-correlated values for the shape's radius. Specifically, we define the base model r_0 as

$$r_0(\theta) = f_0(\theta) + \epsilon_0(\theta), \tag{12}$$

where $\theta \in \mathbb{R}^d$ is a vector containing d equally spaced samples of the interval $[0, 2\pi]$ and $f_0(\theta)$ is the mean radius function. For all experiments, we use $d = 100$ and $f(\theta) = 0.5$.

To add randomness to the mean shape, we use Gaussian Processes (GP), defined by a mean and an exponentiated quadratic kernel

$$k_{mid}(\theta_i, \theta_j) = \sigma_{mid}^2 \exp\left(-\frac{(g(\theta_i) - g(\theta_j))^2}{2l_{mid}^2}\right), \tag{13}$$

where $\theta_i, \theta_j \in \theta$, $g : \mathbb{R} \to \mathbb{R}$ is a function that transforms the domain and mid can be zero or one depending on the stochastic model in Eq. 14 we are discussing.

We define $\epsilon_0(\theta)$ in Equation 12 as the sum of two zero-mean GPs with $g = \sin$ and $g = \cos$ in Eq. 13, respectively. Using these periodic functions ensures that the start and end of the θ interval are mapped to the same radius. The kernel's parameters σ_0 and l_0 define the shape of the contour by affecting the amplitude and the frequency of the angle-correlated noise. We set these parameters to $\sigma_0 = 0.003$ and $l_0 = 0.9$.

To obtain a binary mask from the zero-centered shape defined by the polar coordinates (θ, $r(\theta)$), we first convert them to Cartesian coordinates with the mappings $y = r\sin(\theta)$ and $x = r\cos(\theta)$. Then, we use scikit-image's polygon2mask code to rasterize the resulting closed polygon in a square grid with the target resolution of 300×300 pixels. The panel in the upper left corner of Figure 5 shows a $N = 100$ ensemble generated by sampling the base model r_0 (D1).

For the experiments, we define five datasets of contour ensembles (D2-D6 in Figure 5) based on the three types of outliers we describe next. In all cases, we obtain an outlier-contaminated ensemble by sampling from the mixture

$$r(\theta) = r_0(\theta) + \rho r_1(\theta), \tag{14}$$

where $\rho \sim Bern(0.1)$ introduces an outlier with a probability of 0.1 and r_1 is defined analogously to r_0 in Equation 12. In the following, we describe how we model different outlier types by modifying r_1.

First, we consider magnitude outliers in which we alter the shape's mean radius. We define the auxiliary random variable $sign = 2\gamma - 1$ where $\gamma \sim Bern(0.5)$. $sign$ indicates whether the magnitude contamination corresponds to shrinking (-1) or enlarging (1) the shape. The first dataset with magnitude outliers is the Symmetric Magnitude Contamination (D2) for which $f_1(\theta) = 0.3 \cdot sign$. We define a second dataset with magnitude outliers which we call Peaks Magnitude Contamination (D3). Instead of changing the magnitude of the shape's radius, in D3 we only contaminate a subinterval (θ_l, θ_r) of θ where $\theta_l < \theta_r$ and both θ_l and θ_r are uniformly distributed random variables. Specifically, for D3, we define f_1 as

$$f_1(\theta) = \begin{cases} sign \cdot inc & \theta_l \leq \theta \leq \theta_r \\ 0 & \text{otherwise} \end{cases}$$

where $inc = 0.3$, and θ_l and θ_r are defined for every $\theta_i \in \theta$.

The second type of outlier we consider is shape outliers. To obtain shape outliers, instead of altering the mean radius of the circular shape, we modify the parameters of the covariance matrix of ϵ_1 which define the amplitude (σ_1) and the frequency (l_1) of the noise along the shape's boundary. Specifically, increasing σ_1 leads to higher amplitude while increasing l_1 increases the number of peaks. For the Shape Inside (D4) dataset, we keep $\sigma_1 = 0.003$ but decrease the frequency to $l_1 = 0.01$ to ensure that the shape varies while staying within the ensemble's envelope. For the Shape Outside (D5) dataset, we set $\sigma_1 = 0.009$ and $l_1 = 0.04$, which results in highly varying shapes that spill outside the bounds defined by the normal members of the ensemble. We expect D4 outliers to be more challenging to detect than D5 ones, given that they fall inside the ensemble's envelope.

The final type of outlier we consider are topological outliers which correspond to contours that have holes or disconnected components not present in other members of the ensemble. To create the Different Topologies dataset (D6), we randomly downscale r_1 using a uniform distribution between 0.1 and 0.2 for the scaling factor. Note that we use the same parameters for r_1 as for r_0. After determining the (x, y) coordinates of the shrank shape, we translate them to a random location that lies either inside or outside (with equal probability) of the mean circular shape defined by r_0.

For the experiments, we consider several ensemble sizes $N \in \{i * 10 : 1 \leq i \leq i_{\max}$, where $i_{\max} = 10$ for CBD and $i_{\max} = 30$ for ID. We compute 10 realizations of each dataset/size/depth method combination to establish the statistical significance of the results. We ran all the experiments presented in this section on a Mac Book Pro (2022) with an M1 Pro processor (without GPU acceleration) and 32 GB RAM.

B. Experiment 1: Scaling Behavior

Figure 6 depicts the time in seconds that each depth method takes for ensembles of different sizes. For each size, we compute the mean and standard deviation across replications and datasets (D1-6). The first thing to note is that we only ran CBD methods until $N = 100$. After this point, the CBD method took too long to compute. In contrast, we considered ensembles up to size $N = 300$ for ID. The figure shows how ID and eID, with a computational complexity of $\mathcal{O}(N^2)$, scale more favorably than CBD methods, which are $\mathcal{O}(N^3)$.

In addition to the aggregated runtime, we investigated the time the preprocessing and depth calculation loop portions of each method take. Table I shows this information for D1 with $N = 100$. As the table shows, all methods spend most of their time in the depth calculation loop (t2). CBD methods take, on average, an order of magnitude more time than ID methods. The large standard deviations of CBD methods' timings are

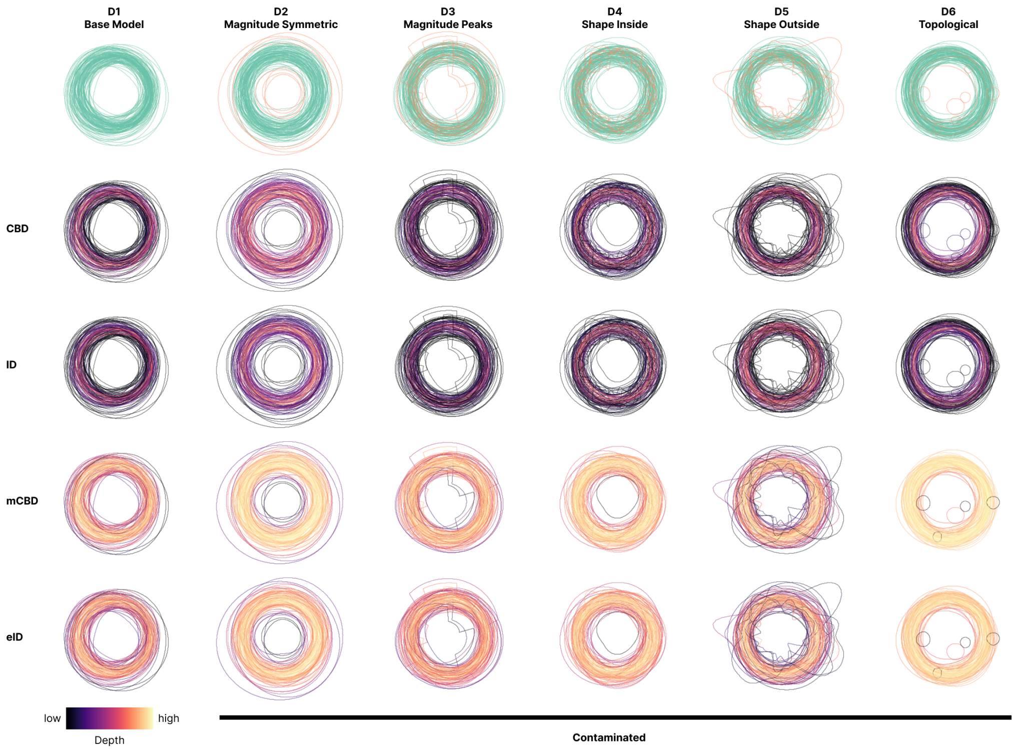


Fig. 5. The first row presents an overview of the synthetic datasets we used in the experiments, with the outliers highlighted in orange. The last four rows plot the ensembles assigning the lines' colors based on the depths each method yielded. Darker and brighter colors denote lower and higher depth values, respectively. The color scale was scaled based on the min and max depth value per dataset/depth method combination to facilitate the comparison of the depth-induced rankings across methods.

TABLE I
MEAN AND STANDARD DEVIATION OF THE PREPROCESSING (T1), DEPTH CALCULATION LOOP (T2) AND FULL (T3=T1+T2) TIMES IN SECONDS FOR D1 WITH $N = 100$.

Method	t1 (secs)	t2 (secs)	t3 (secs)
CBD	6.75 ± 1.77	612.31 ± 351.40	619.06 ± 351.14
mCBD	6.48 ± 1.46	697.02 ± 328.91	703.50 ± 328.49
ID	0.00 ± 0.00	2.31 ± 0.37	2.31 ± 0.37
eID	0.00 ± 0.00	7.37 ± 3.98	7.37 ± 3.98

caused by outlier timings that arose likely due to other processes in the machine interfering with the experiment's process. Within each method family, the modified version takes more time because they require more operations than the strict versions. Finally, CBD methods have a larger preprocessing time (t1) than ID methods, which do not require preprocessing. This is specific to our implementation, which precomputes CBD's bands before starting the depth calculation loop.

C. Experiment 2: Outlier Detection

Depths can be used to perform robust statistical analysis by removing outliers, which are contours with low depth. For the second experiment, we evaluate ID's performance in identifying outliers in D2-D6 in Fig. 5. Specifically, given a set of outliers $\mathcal{O}_m$ for a method m and a reference set $\mathcal{O}_r$, we compute the percentage of correctly identified outliers with respect to the reference set as

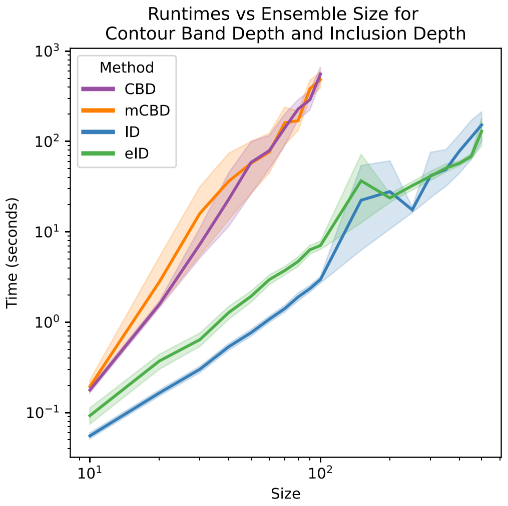


Fig. 6. Comparison of mean runtimes across datasets and replications of CBD, mCBD, ID and eID. Both x and y-axis use logarithmic scales and shaded area denotes the 95 percent confidence interval.

$$PO_{m,r} = \begin{cases} 0 & \text{if } |O_r| = 0 \\ \frac{|O_m \cap O_r|}{|O_r|} & \text{otherwise,} \end{cases} \tag{15}$$

where $|\cdot|$ denotes the number of outliers in the set.

TABLE II

Average percentage of outliers that Contour Band Depth (CBD) and Inclusion Depth (ID) methods detected with respect to the ground truth outliers for $N = 100$.

Dataset	CBD (%)	mCBD (%)	ID (%)	eID (%)
D2	76.16 ± 13.31	98.12 ± 4.22	90.08 ± 7.68	98.12 ± 4.22
D3	77.54 ± 14.67	58.94 ± 13.19	71.46 ± 14.48	49.89 ± 14.20
D4	88.14 ± 15.59	17.43 ± 14.74	85.07 ± 14.34	8.06 ± 7.68
D5	85.21 ± 16.92	69.27 ± 8.94	83.37 ± 18.90	54.54 ± 7.78
D6	66.11 ± 9.31	68.19 ± 16.23	81.46 ± 13.33	66.52 ± 18.90

For a method m, we define its set of outliers O_m as the $\lceil N\alpha \rceil$ members with the lowest depths, where $\lceil \cdot \rceil$ is the ceiling operator. For the results we report next, we used $\alpha = 0.3$. We compare the outliers of each depth method identified against the ground truth (GT) outliers, which we define as the reference set O_r. Table II shows the mean and the standard deviation of the percentage of the outliers each method detected with respect to the GT ones for D2-D6 with $N = 100$.

As the table indicates, except for D2, strict depth methods are more effective at identifying outliers. This result agrees with the functional depth literature, which shows that strict depth methods have a higher sensitivity to outliers [13]. The most challenging dataset for mCBD and eID was D4, with inside-shape outliers. Although both methods performed poorly, mCBD did a better job, which potentially indicates that the extra comparisons of CBD endow the method with a higher sensitivity for detecting shape outliers.

As the table indicates, no strict method consistently outperforms the other. ID performed better for the dataset with symmetric magnitude contamination (D2) and topological outliers (D6). In the other cases, CBD achieved better scores. Similarly, except for D4, the performance of modified depth methods was comparable across datasets. These results show how, in practice, the choice of method will depend on the type of data at hand. In agreement with previous literature in band depths [7], [13], the strength of CBD lies in identifying outliers like those in D4, which have a significantly different shape but fall within the ensemble's band envelope.

Finally, we also compare the methods' outlier detection performance qualitatively. The four bottom rows of Figure 5 present the spaghetti plots with lines colored according to the depths that different methods yield. The figure evidences the similarities between CBD and ID, and mCBD and eID. CBD and ID produce a wider range of depth values, demonstrated by the color gradient which contains black and bright yellow lines. In contrast, mCBD and eID yield mostly high-depth scores with some contours receiving lower ones. Graphically, this translates to overall brighter color gradients. Despite this visual change, it is possible to observe that in most cases, the depth-induced rankings of the contours are similar between strict and modified versions.

D. Experiment 3: Estimator's Robustness

Depth values permit generalizing uni-variate order and rank statistics to the multivariate case. For this experiment, we are interested in the quality of the trimmed mean, which is one of the robust statistics that the contour boxplot visualization uses. To compute the α-trimmed mean (M_m^α) of an ensemble of contours we average binary masks of the top $N - \lceil N\alpha \rceil$ contours, depth-wise, and extract a new contour from the resulting scalar field using 0.5 as iso-value. Specifically, we compute the α-trimmed mean contour for method m using the expression

$$\mathrm{M}_m^\alpha = \frac{\sum_{i=1}^{N-\lceil N\alpha \rceil} in(c_i)}{N - \lceil N\alpha \rceil}, \tag{16}$$

where $in(c_1), ..., in(c_{N-\lceil N\alpha \rceil})$ are the binary masks of the inside regions associated with the $N - \lceil N\alpha \rceil$ contours with the highest depth, according to method m. In addition to each method's trimmed mean, we also consider the sample mean (M_S), which we compute per dataset/replication combination by using Eq. 16 without trimming the ensemble. M_S represents a worst-case scenario in which outliers were not removed. For the experiments in this section, we set $\alpha = 0.3$.

A robust trimmed mean is one not affected by outliers. In other words, the trimmed mean contour should be close to the population's average shape. Therefore, to evaluate the depth methods' estimators, we compare them against the binary mask of f_0 in Eq. 12, which we denote M_P. To compare the trimmed means with M_P we compute the mean squared error (MSE) between the masks

$$MSE(M_m^\alpha, M_P^\alpha) = \frac{\sum_{r=0}^{rows} \sum_{c=0}^{cols} [M_m^\alpha(r,c) - M_P^\alpha(r,c)]^2}{rows \times cols}, \tag{17}$$

where $M_m^\alpha(r,c)$ is the value of the binary array of the trimmed mean M_i^α under consideration at the given row and column.

Table III presents the mean and the standard deviation of the MSE for D1-D6 with the ensemble size $N = 100$. Both CBD and ID methods yield lower average MSE when compared to the sample mean M_S. This shows that removing outliers, only considering the most central contours, leads to more robust estimators closer to the population mean M_P. In most cases, the mean MSE of M_α^{CBD} is higher than that of M_α^{ID}. The same observation holds for the modified versions, which suggests that the outliers ID methods remove contribute more to deviating the trimmed mean from the population estimate. Finally, modified depth methods obtain lower MSE than their strict counterparts. Considering that strict methods performed better at identifying outliers, this result suggests that other contours besides artificially introduced outliers might contribute more towards making the mean estimates less robust. These results show that both CBD and ID methods yield robust mean estimates that are closer to the population estimate than M_S.

VII. Visual Comparison on Real Data

The previous results demonstrated ID's robustness and more favorable scaling behavior compared to CBD using synthetic data. We now illustrate the use of ID with medical image semantic segmentation and meteorological forecasting datasets. The contours in these real datasets tend to cross over a lot. Therefore, we focus the analysis on eID, which yields more visually meaningful results in these cases. Unless stated otherwise, we used the same setup for the depth computation

TABLE III
AVERAGE MSE BETWEEN POPULATION ESTIMATE M_P, AND THE SAMPLE MEAN (M_S) AND ALPHA-TRIMMED MEANS OBTAINED WITH CBD (M^{α}_{CBD}), MCBD (M^{α}_{mCBD}), ID (M^{α}_{ID}) AND EID (M^{α}_{eID}) DEPTHS. WE COMPUTE THE AVERAGE MSE ACROSS REPLICATIONS FOR $N = 100$ AND INCLUDE ALSO THE STANDARD DEVIATION OF THE ESTIMATES. WE MULTIPLY BOTH THE MEAN AND STD BY $\times 10^2$ TO REDUCE CLUTTER.

Dataset	M_S	M^{α}_{CBD}	M^{α}_{mCBD}	M^{α}_{ID}	M^{α}_{eID}
D1	1.42 ± 0.06	1.17 ± 0.10	1.13 ± 0.04	1.15 ± 0.08	1.12 ± 0.05
D2	1.77 ± 0.08	1.47 ± 0.12	1.32 ± 0.14	1.37 ± 0.14	1.31 ± 0.12
D3	1.51 ± 0.08	1.26 ± 0.11	1.20 ± 0.12	1.24 ± 0.10	1.18 ± 0.11
D4	1.46 ± 0.08	1.24 ± 0.09	1.14 ± 0.05	1.22 ± 0.08	1.13 ± 0.05
D5	1.50 ± 0.08	1.24 ± 0.10	1.17 ± 0.07	1.24 ± 0.10	1.17 ± 0.07
D6	1.60 ± 0.16	1.48 ± 0.23	1.17 ± 0.08	1.24 ± 0.14	1.15 ± 0.06

methods and ran the analyses in the same machine as in the experiments with synthetic data.

A. Medical Image Segmentation Ensembles

1) Data: In image-guided medical specialties, clinicians use three-dimensional images of the patient's anatomy to plan the treatment. A core step of the treatment planning process is to segment anatomies of interest like malignancies and the organs-at-risk. With the advent of deep learning-based auto-contouring technologies, this step has been largely automated [29]. Nevertheless, clinicians still need to perform a quality assessment of the segmentations, which requires understanding the uncertainty in the predictions.

We consider the computerized tomography (CT) of a patient with head and neck cancer treated at HollandPTC between 2018 and 2020. The IRB approved the research protocol for the use of patient data in research, all patients signed an informed consent form. For the analysis, we focus on the brain stem and the parotid gland because these structures are not always clearly visible in CT, which can increase inter-clinician variability. In these cases, a visual statistical summary can help clinicians understand the range of predictions. We used a collection of 3D segmentation models based on the popular UNet architecture [30] to generate an ensemble of segmentation predictions of the right parotid gland. Specifically, we trained 30 models on different subsets of the training split of the dataset of the Head and Neck Auto Segmentation MICCAI Challenge [31], a technique known as bootstrapping in the machine learning community. The MICCAI dataset contains CT scans of patients with head and neck cancer with ground truth segmentations of nine organs at risk. To further augment the ensemble size, and the variability of the predictions, we trained each model using different learnable weight initializations. Using the resulting models to segment the parotid gland yields an ensemble of 120 scalar maps of per-voxel softmax probabilities. We extracted the contour ensemble that CBD and ID receive as input by thresholding these arrays with an iso-value of 0.8. For the results below, we computed the depths of the ensemble of contours in 2D 540 × 540 pixels slices of the right parotid gland and brain stem segmentation volumes.

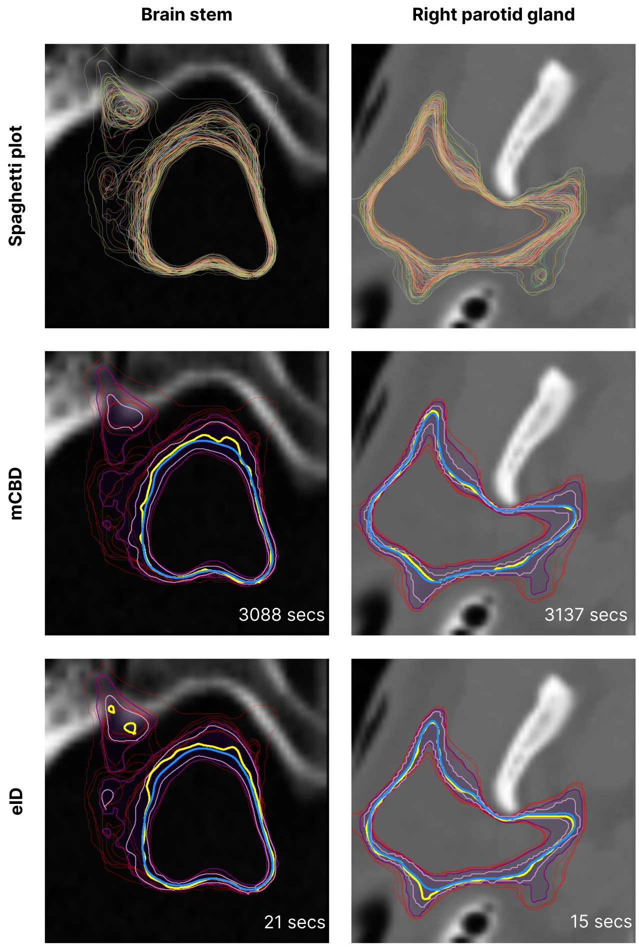


Fig. 7. Contour boxplots that provide a statistical summary of an ensemble of contours of a slice of the brain stem (top row) and right parotid gland (bottom row) of a head-and-neck cancer patient. We generated the contour boxplots using the depths obtained from the mCBD and eID. The yellow and blue lines correspond to the median and mean, respectively. Two bands are depicted in shades of purple as formed by members with the top 50% and 100% depths, not considering outliers, which are shown using dashed red lines.

2) Analysis: The top row of Figure 7 visualizes the raw ensemble of contours of the brain stem and parotid gland using spaghetti plots. The variability in the contours of the two structures differs. The brain stem shows significantly more variability than the parotid gland, especially on the upper left side where several contour lines go out of the way of the main shape. One possible reason for this behavior is that the brain stem is harder to segment on a CT scan because of the lack of contrast and landmarks. In these cases, clinicians usually require additional image modalities like magnetic resonance. In contrast, even though the parotid gland is also made of soft tissue, it is surrounded by bone tissue and visible landmarks like the patient's face, which makes it slightly easier to segment using CT alone. Nevertheless, similar to the brain stem, in the lower right part of the parotid gland, where there is less contrast, it is possible to observe how the model produced more variable results.

Visual statistical summaries remove the need from presenting all ensemble members while still conveying relevant statistical features like the representative contours and the ensemble's variability. For each anatomical structure, Figure 7

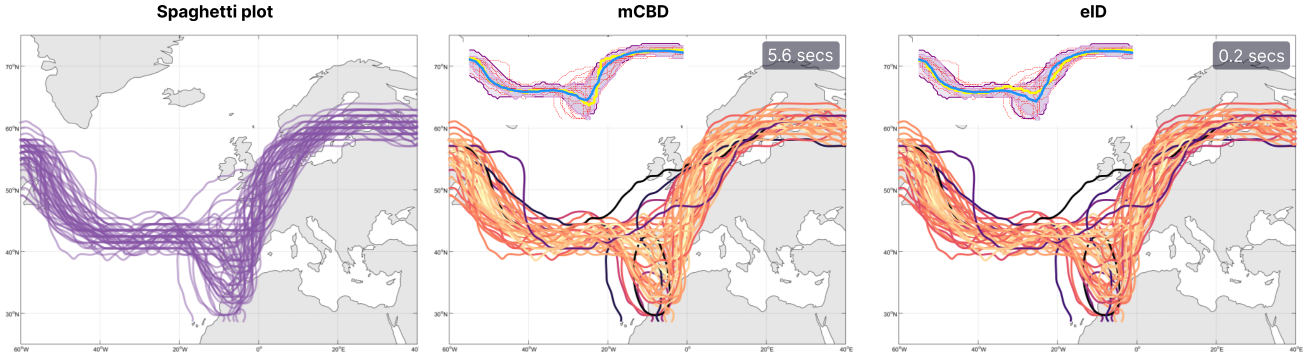


Fig. 8. mCBD and eID depths for an ensemble of 500 hPa geopotential height contour lines. The inset of each method presents the corresponding contour boxplot with the $N \times 20\% = 10$ contours with the lowest depth set as outliers.

presents contour boxplots generated with depths from the mCBD and eID, using $\alpha = 0.1$ for the trimming. The first thing to note is the different runtimes. For a $N = 100$ ensemble, mCBD took more than twenty minutes to compute the depths. In contrast, it took eID seconds. These results show that ID can support larger datasets without requiring special hardware, which increases its practical value.

In terms of the boxplot's statistical features, we start by analyzing the median, depicted as a yellow line. In both cases, the median that mCBD and eID yield is not the same contour. Nevertheless, the contours' shapes are visually similar. When we inspected the raw data, we noticed both medians obtained high depth with both methods, but their ranks varied, which resulted in a different contour being displayed. The similarity of the rankings induced by mCBD and eID depths can be observed by comparing the method's trimmed means (blue lines). The figure shows how the means are similar, which indicates stable rankings of the contours. Finally, the figure shows that the three methods identified the most visually deviant contours of the brain stem and the parotid gland as outliers, which is another reason for the stability of the α-trimmed mean across methods.

B. *Meteorological Forecasting*

1) Data: A common use case for contour statistical models is to analyze meteorological forecast data. In this work, we consider data from the European Centre for Medium-Range Weather Forecasts (ECMWF). Specifically, the ECMWF Ensemble Prediction System (ENS) provides ensembles of predictions for different variables like precipitation, temperature, and pressure. The forecasts include $N = 50$ perturbed members and a control run. We analyze the same data as in [5], which is the forecast from 00:00 UTC 15 October 2012. More details about this type of data can be found at [32]. The region under consideration encompasses $101 \times 41 \times 62$ grid points, which corresponds to latitude, longitude, and geopotential height dimensions. For the analysis, we consider 2D fields, corresponding slices of the region where the geopotential height is $500hPa$. To obtain contours from this field, we threshold them using an iso-value of 5600 m. The left-most panel of Figure 8, depicts the extracted contours laid over the geographical region they span.

2) Analysis: Without any coloring or grouping of the lines in the first column of Figure 8 it is hard to discern patterns and main trends in the ensemble. The second and third panels of Figure 8 color the lines using the depth that mCBD and eID assigned to each contour. Darker and brighter colors represent lower and higher values, respectively. We scaled the color scale based on the min and max depth each method yielded. It is possible to observe how mCBD and eID yield similar depth scores, evident when comparing the color gradients. Nevertheless, similarly to the case of segmentation data, the specific depth values vary, altering the depth-induced rankings. The insets in the second and third panels show how different rankings lead to different medians (yellow lines) being displayed. Despite the median varying, the 0.1-trimmed mean (blue line) and outliers (red dashed lines) show that the outlier sets (dashed red lines) of mCBD and eID largely overlap, which leads to robust mean estimates that are visually similar.

VIII. Discussion and Conclusion

In this paper, we presented Inclusion Depth (ID), a new depth notion applicable to contour ensembles. The concept of statistical depth permits extending order and rank-based statistics to the multivariate case. Depth-induced orderings allow summarizing the ensemble members in terms of their median, trimmed mean, and confidence bands, and obtaining robust estimators by removing outliers.

ID provides theoretical guarantees on the depth estimates, derived from its relationship with Half-Region Depth. Additionally, based on the simple principle of assessing contours inside/outside relationships, ID is accessible and its results interpretable. Using synthetic data we demonstrated ID's more favorable $\mathcal{O}(N^2)$ scaling, compared CBD's $\mathcal{O}(N^3)$ [7]. Furthermore, the experiments showed that ID and eID are successful at identifying a wide range of outliers and yield robust estimators of the ensemble's mean, comparable to CBD's. These robust estimators enable extending robust statistical theory and analysis to contours. Finally, by applying ID to real datasets, we further demonstrated the method's practical value to analyze contour ensembles when paired with visualizations like contour boxplots.

In the literature, it has been noted that CBD can be accelerated in several ways. First, CBD's outer loop is highly parallelizable, so it could significantly profit from GPU acceleration. In this paper, we did not focus on improvements that could be added on top of the methods. Rather, we propose an alternative depth notion that is asymptotically faster than CBD. Similarly to CBD, ID has a highly parallelizable loop, so this improvement would also benefit ID. Second, in terms of algorithmic improvements, [33] proposes a faster way to compute functional Band Depth. Contours, with the different possible topologies, are not straightforward to adapt to this methodology. Therefore, it remains future work to verify whether these optimizations are possible. Same as with parallelization, it holds that such an improvement would likely benefit both CBD and ID.

The experiments with synthetic data showed that ID and eID detect outliers with comparable performance to CBD across several outlier types. Nevertheless, there is still room for improvement. Particularly in the case of eID, which performed weakly at identifying shape outliers with a magnitude similar to other ensemble members. Improving outlying detection performance constitutes future work. We anticipate that introducing information about the contour's metric structure, similar to [6], could help in cases where inside/outside relationships do not suffice. Second, the eID can assign low non-zero depth scores to outlying contours. mCBD uses an automatic thresholding method that optimizes the ensemble's mean depth to set outliers' depth to zero. This procedure removes the need to find a threshold for the trimming operations via trial and error, like in eID's case. To reduce users' burden, we will investigate options to integrate an automated thresholding procedure similar to mCBD's in our framework.

The improved computational complexity of ID brings within reach the usage of depth-based order and rank statistics for larger datasets in interactive settings. In domains like computer-aided design, simulation, and medical image segmentation, it is common to deal with three-dimensional objects [11]. Our method is quite general and can be applied to three-dimensional contours with ease. Second, currently unimodal distribution is assumed, however, when studying contour's ensembles it is common to first identify the main modes of variation [4], [19], [34]. CBD could make this identification more robust to certain types of outliers [15] but at the cost of reduced interactivity. Using ID instead would permit performing real-time interactive depth-based clustering on larger contour ensembles. Finally, the interactivity that ID unlocks calls for reimagining contour boxplots for interactive scenarios. For instance, it could be possible to change parameters or weights in the depth function and see them reflected in the contour boxplot in real time.

Acknowledgments

The research for this work was funded by Varian, a Siemens Healthineers Company, through the HollandPTC-Varian Consortium (grant id 2019022), and partly financed by the Surcharge for Top Consortia for Knowledge and Innovation (TKIs) from the Ministry of Economic Affairs and Climate. We also want to thank Rüdiger Westermann and Keving Hölhein for sharing with us the meteorological forecasting data from the ECMWF.

References

[1] D. B. Stephenson and F. J. Dolas-Reyes, "Statistical methods for interpreting monte carlo ensemble forecasts," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 52, no. 3, pp. 300–322, 2000. [Online]. Available: https://doi.org/10.3402/tellusa.v52i3.12267

[2] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001081

[3] F. Renard, S. Guedria, N. D. Palma, and N. Vuillerme, "Variability and reproducibility in deep learning for medical image segmentation," *Scientific Reports*, vol. 10, no. 1, p. 13724, 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-69920-0

[4] M. Zhang, Q. Li, L. Chen, X. Yuan, and J.-H. Yong, "Enconvis: A unified framework for ensemble contour visualization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[5] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Visual analysis of spatial variability and global correlations in ensembles of iso-contours," *Computer Graphics Forum*, vol. 35, no. 3, pp. 221–230, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12898

[6] I. Demir, M. Jarema, and R. Westermann, "Visualizing the central tendency of ensembles of shapes," in *SIGGRAPH ASIA 2016 Symposium on Visualization*, ser. SA '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/3002151.3002165

[7] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2713–2722, 2013.

[8] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. Röber, and R. Westermann, "Visualization in meteorology—a survey of techniques and tools for data analysis tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 12, pp. 3268–3296, 2018.

[9] M. Mirzargar and R. T. Whitaker, "Representative consensus from limited-size ensembles," *Computer Graphics Forum*, vol. 37, no. 3, pp. 13–22, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13397

[10] P. Voglreiter, P. Mariappan, M. Pollari, R. Flanagan, R. Blanco Sequeiros, R. H. Portugaller, J. Fütterer, D. Schmalstieg, M. Kolesnik, and M. Moche, "Rfa guardian: Comprehensive simulation of radiofrequency ablation treatment of liver tumors," *Scientific Reports*, vol. 8, no. 1, p. 787, 2018. [Online]. Available: https://doi.org/10.1038/s41598-017-18899-2

[11] M. Raj, M. Mirzargar, J. S. Preston, R. M. Kirby, and R. T. Whitaker, "Evaluating shape alignment via ensemble visualization," *IEEE Computer Graphics and Applications*, vol. 36, no. 3, pp. 60–71, 2016.

[12] K. Mosler, *Depth Statistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 17–34. [Online]. Available: https://doi.org/10.1007/978-3-642-35494-6_2

[13] S. López-Pintado and J. Romo, "On the concept of depth for functional data," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 718–734, 2009. [Online]. Available: https://doi.org/10.1198/jasa.2009.0108

[14] C. Gao, "Robust regression via mutivariate regression depth," *Bernoulli*, vol. 26, no. 2, pp. 1139 – 1170, 2020. [Online]. Available: https://doi.org/10.3150/19-BEJ1144

[15] R. Jörnsten, "Clustering and classification based on the l1 data depth," *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 67–89, 2004, special Issue on Multivariate Methods in Genomic Data Analysis. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0047259X04000272

[16] J. Wang, S. Hazarika, C. Li, and H.-W. Shen, "Visualization and visual analysis of ensemble data: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 9, pp. 2853–2872, 2019.

[17] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead, "Noodles: A tool for visualization of numerical weather model ensemble uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1421–1430, 2010.

[18] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus, "Visualizing confidence in cluster-based ensemble

weather forecast analyses," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 109–119, 2018.

[19] F. Ferstl, K. Bürger, and R. Westermann, "Streamline variability plots for characterizing the uncertainty in vector field ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 767–776, 2016.

[20] K. Pothkow and H.-C. Hege, "Positional uncertainty of isocontours: Condition analysis and probabilistic measures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1393–1406, 2011.

[21] T. Athawale, E. Sakhaee, and A. Entezari, "Isosurface visualization of data with nonparametric models for uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 777–786, 2016.

[22] K. Pöthkow and H.-C. Hege, "Nonparametric models for uncertainty visualization," *Computer Graphics Forum*, vol. 32, no. 3pt2, pp. 131–140, 2013.

[23] S. López-Pintado and J. Romo, "A half-region depth for functional data," *Computational Statistics & Data Analysis*, vol. 55, no. 4, pp. 1679–1695, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947310004123

[24] R. Y. Liu, "On a notion of data depth based on random simplices," *The Annals of Statistics*, vol. 18, no. 1, pp. 405–414, 1990. [Online]. Available: http://www.jstor.org/stable/2241550

[25] J. W. Tukey, "Mathematics and the picturing of data," *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, vol. 2, pp. 523–531, 1975. [Online]. Available: https://cir.nii.ac.jp/crid/1573950399770196096

[26] R. Serfling and Y. Zuo, "General notions of statistical depth function," *The Annals of Statistics*, vol. 28, no. 2, pp. 461 – 482, 2000. [Online]. Available: https://doi.org/10.1214/aos/1016218226

[27] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. 20, no. 2, pp. 316–334, 2011. [Online]. Available: https://doi.org/10.1198/jcgs.2011.09224

[28] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[29] E. Montagnon, M. Cerny, A. Cadrin-Chênevert, V. Hamilton, T. Derennes, A. Ilinca, F. Vandenbroucke-Menu, S. Turcotte, S. Kadoury, and A. Tang, "Deep learning workflow in radiology: a primer," *Insights into Imaging*, vol. 11, no. 1, p. 22, 2020. [Online]. Available: https://doi.org/10.1186/s13244-019-0832-5

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

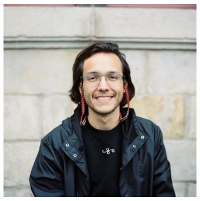
[31] P. F. Raudaschl, P. Zaffino, G. C. Sharp, M. F. Spadea, A. Chen, B. M. Dawant, T. Albrecht, T. Gass, C. Langguth, M. Lüthi, F. Jung, O. Knapp, S. Wesarg, R. Mannion-Haworth, M. Bowes, A. Ashman, G. Guillard, A. Brett, G. Vincent, M. Orbes-Arteaga, D. Cárdenas-Peña, G. Castellanos-Dominguez, N. Aghdasi, Y. Li, A. Berens, K. Moe, B. Hannaford, R. Schubert, and K. D. Fritscher, "Evaluation of segmentation methods on head and neck ct: Auto-segmentation challenge 2015," *Medical Physics*, vol. 44, no. 5, pp. 2020–2036, 2017. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12197

[32] M. Leutbecher and T. Palmer, "Ensemble forecasting," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3515–3539, 2008.

[33] Y. Sun, M. G. Genton, and D. W. Nychka, "Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked?" *Stat*, vol. 1, no. 1, pp. 68–74, 2012.

[34] B. Ma and A. Entezari, "An interactive framework for visualization of weather forecast ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1091–1101, 2019.

IX. Biography Section



Nicolas F. Chaves-de-Plaza is a PhD student at the Department of Intelligent Systems at the Delft University of Technology. He works on developing visualization and interaction tools to support clinician-driven segmentation of 3D medical images in the context of Adaptive Proton Therapy.

Prerak Mody is a PhD student at the Division of Image Processing (Dutch abbreviation LKEB) at Leiden University Medical Center. His research focuses on clinically-applicable bayesian and interactive deep learning techniques.

Marius Staring is a professor and vice director of LKEB at the Leiden University Medical Center. He and his team develop generic machine-learning approaches for automated image analysis and apply these in the clinical and life sciences. He is an Associate Editor of IEEE TMI, and a member of program committees of MICCAI, IEEE ISBI, SPIE MI, and WBIR. Open-sourcing his methods has been a common theme in his career, exemplified by the image registration package Elastix, see https://elastix.lumc.nl/.

René van Egmond is an associate professor of Cognitive Ergonomics at the Faculty of Industrial Design Engineering at the Delft University of Technology. His expertise lies in the fields of Product Sound Design & Perception and Informational Ergonomics. His research is focused on understanding how people process information streams in complex environments and how people process this information and deal with this complexity.

Anna Vilanova is a full professor in visual analytics at the Department of Mathematics and Computer Science at Eindhoven University of Technology. She is leading a research group on the subject of visual analytics focusing on high dimensional data, explainable AI, and medical visualization for complex imaging data. She has been a member of international program committees, chair, and editor of conferences & journals in visualization. She has been an elected member of the EUROGRAPHICS executive committee and vice president of EUROGRAPHICS.

Klaus Hildebrandt is a tenured Assistant Professor at the Computer Graphics and Visualization Group at the Department of Intelligent Systems at Delft University of Technology. His research interests include Visual Computing, Geometric Data Processing, Physical Simulation, and Computational and Discrete Differential Geometry.