SING: Stability-Incorporated Neighborhood Graph

Diana Marin, Amal Dev Parakkat, Stefan Ohrhallinger, Michael Wimmer, Steve Oudot, Pooran Memari

SING: Stability-Incorporated Neighborhood Graph

DIANA MARIN, Technische Universität Wien, Austria
AMAL DEV PARAKKAT, LTCI - Télécom Paris, Institut Polytechnique de Paris, France
STEFAN OHRHALLINGER, Technische Universität Wien, Austria
MICHAEL WIMMER, Technische Universität Wien, Austria
STEVE OUDOT, Inria and École Polytechnique, France
POORAN MEMARI, CNRS, LIX, École Polytechnique, Inria, France

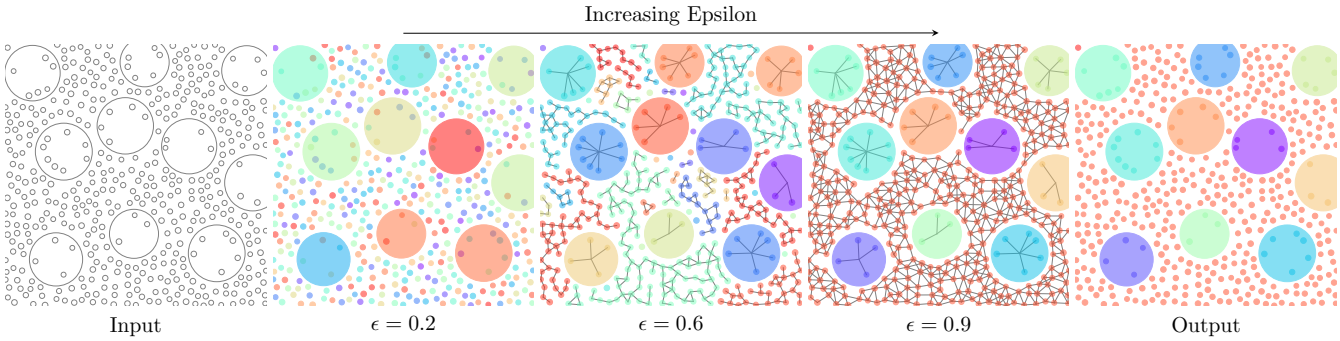


Fig. 1. Left - a disk distribution input; middle - SING results, utilizing a disk distance from [Ecormier-Nocca et al. 2019], without any prior class information nor post-processing. Increasing ε (left to right) yields nested proximity graphs and diverse clustering effects, represented in random colors; right - output clusters.

We introduce the Stability-Incorporated Neighborhood Graph (SING), a novel density-aware structure designed to capture the intrinsic geometric properties of a point set. We improve upon the spheres-of-influence graph by incorporating additional features to offer more flexibility and control in encoding proximity information and capturing local density variations. Through persistence analysis on our proximity graph, we propose a new clustering technique and explore additional variants incorporating extra features for the proximity criterion. Alongside the detailed analysis and comparison to evaluate its performance on various datasets, our experiments demonstrate that the proposed method can effectively extract meaningful clusters from diverse datasets with variations in density and correlation. Our application scenarios underscore the advantages of the proposed graph over classical neighborhood graphs, particularly in terms of parameter tuning.

CCS Concepts: • **Computing methodologies** → **Point-based models**; • **Mathematics of computing** → **Algebraic topology**.

Additional Key Words and Phrases: Proximity graphs, clustering, persistence analysis, K-means, Rips complexes, Neighborhood graph, topological data analysis, point patterns, similarity metric, discrete distributions, Stipple art editing, Pattern design, Network topology

Authors' Contact Information: Diana Marin, Technische Universität Wien, Wien, Austria, dmarin@cg.tuwien.ac.at; Amal Dev Parakkat, LTCI - Télécom Paris, Institut Polytechnique de Paris, Paris, France, amal.parakkat@telecom-paris.fr; Stefan Ohrhallinger, Technische Universität Wien, Wien, Austria, ohrhallinger@cg.tuwien.ac.at; Michael Wimmer, Technische Universität Wien, Wien, Austria, wimmer@cg.tuwien.ac.at; Steve Oudot, Inria and École Polytechnique, Paris, France, steve.oudot@inria.fr; Pooran Memari, CNRS, LIX, École Polytechnique, Inria, Paris, France, memari@lix.polytechnique.fr.

1 Introduction

Context and motivation. Clustering serves as a fundamental algorithmic procedure in data analysis, extensively employed in extremely diverse fields such as biology, astronomy, art, medicine, as well as computer graphics and vision. Despite these important applications, existing clustering algorithms suffer from a variety of drawbacks, and no universal solution has emerged. In this work, we propose an intuitive proximity criterion, leading to a stable and efficient clustering algorithm which consistently achieves accurate grouping across diverse data types. Experimental results demonstrate that our method, characterized by its simplicity and elegance, matches or exceeds the performance of state-of-the-art clustering algorithms across multiple application scenarios. It effectively handles density variations and multi-class data while robustly extending to noisy datasets using stable persistence-based parameter tuning.

Key Idea. The proximity criterion introduced in this paper represents a generalization of the classical neighborhood graph concept. Instead of simply connecting each point to its nearest neighbor, we utilize the distance to the nearest neighbor as a feature to determine its proximity to other points. This approach elegantly incorporates local density information into the proximity measure in a formal manner, offering an intuitive improvement to traditional methods.

Validation via Clustering. Leveraging our proximity measure between element pairs allows for the application of various clustering techniques that depend on element similarities in different manners. For instance, in center-based algorithms like k-means, a small set of potential cluster centers is initialized from the data and iteratively refined. In affinity propagation, data points interact via a graph

structure to select a subset of points as representatives. Our proximity criterion based on local density offers computational efficiency, flexibility in terms of similarity constraints, and stability advantages derived from the persistence analysis approach.

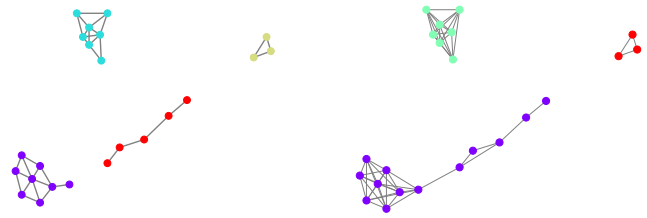
Stability via TDA tools. To better interpret our proximity criterion and to incorporate stability and robustness of the induced clustering with respect to noise on the overall data, we employ a common tool in topological data analysis (TDA): Persistent Homology. Specifically, we focus on order-one homology and the analysis of connected components, which simplifies the process. Persistent diagrams allow us to capture topological features of our proximity graph across different scales, leading to optimal parameter values and offering proven stability in the resulting clusters.

Highlights of Contributions. The proposed proximity graph is characterized by its intuitive simplicity, ease of use, and adaptability to high-dimensional spaces and non-Euclidean metrics. Unlike many existing clustering methods, our SING clustering algorithm does not require prior knowledge of the number of clusters, as it optimizes the clustering outcome based on the stability of the created clusters in terms of parameters and persistence lifespans. Its stability is demonstrated based on the existing stability results of persistent homology. It has remarkable flexibility in terms of the type of distance that can be fed into it, opening up interesting avenues of research for proximity adaptation. This includes consideration for anisotropic metrics, surface curvature, or user-defined local constraint encoding via distance prescription for interactive analysis. We showcase the flexibility of our method via various application scenarios employing different metrics or analytical approaches. While we mainly focus on applications related to stipple clustering, our method finds broad applicability across different domains, such as data segmentation, multi-class disk distribution analysis, shape reconstruction, and network topology analysis. We briefly investigate these tangent directions, and we leave further exploration and research into metric choices as a potential source of inspiration for future work in the CG community. The source code is available online - https://github.com/dianam76/SING.

2 Related work

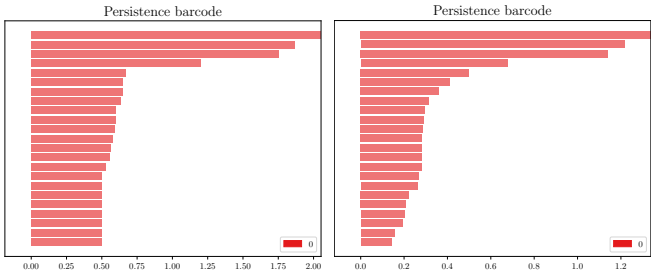
Let us briefly discuss two categories of related work in this context.

Proximity Graphs. A comprehensive overview of neighborhood graphs is given in [Jaromczyk and Toussaint 1992]. The ε-neighborhood graph has the data points as vertices and connects every pair of data points that are within distance ε of each other. The nearest neighbor graph (NN) connects each point to its nearest neighbor, while the minimum spanning tree (MST) connects all components of the NN with minimum length edges. The relative neighborhood graph (RNG) connects two points if no other point is closer to both of them than their mutual distance. The Gabriel graph (GG) connects two points if the smallest closed disk that contains them is empty of other points. The Delaunay triangulation (DT) contains triangles with circumcircles empty of other points. The α-complex is the subset of the DT that can be enclosed in a disc with radius $1/\alpha$ that is empty of other points. The β-skeleton is a scale-invariant version of α-shapes containing those edges which make an angle with another



(a) SING clusters (with $\varepsilon = 0.9$) (b) Rips with distance = 1

Fig. 2. Connected components of SING compared to the Rips complex.



(a) Barcode for SING. (b) Barcode in the Euclidean distance.

Fig. 3. Barcodes of the Rips filtrations for SING and for the ε-neighborhood graph in the ambient Euclidean distance, respectively, on the data of Figure 2. The barcode for SING indicates the possibility of having 3 or 4 clusters, while the other barcode only indicates the possibility of having 3 clusters.

point smaller than β. These proximity graphs are related in the following way: NN $\subseteq$ MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ DT, with RNG and GG as special cases of the β-skeleton family. The γ-neighborhood graph family [Veltkamp 1992] reduces to the β-skeleton, the DT, or to the convex hull, depending on γ. The sphere-of-influence graph (SIG) connects two points if their distance is less or equal to the summed distances to their respective nearest neighbors. In the planar case, the SIG has at most cn edges for n points, with $c \leq 17.5$ [Avis and Horton 1985] and its construction time complexity is $O(n \log n)$ [Bentley and Ottmann 1979]. The k-th SIG extends the sphere to contain k nearest neighbors instead of just one [Guibas et al. 1992]. The SIG was also applied in two parallel works for curve reconstruction from points: [Marin et al. 2022] and [de Figueiredo and Paiva 2022]. In [de Figueiredo and Paiva 2022], a parameterized planar variation of the SIG is introduced, by intersecting it with the Delaunay triangulation, making it suitable for 2D region boundary extraction.

We will present the Stability-Incorporated Neighborhood Graph (SING) as an extension of the SIG graph by incorporating additional features that offer more flexibility and control in encoding proximity information via local density encoding, along with stability results as an inherent property, derived from a complementary TDA analysis.

Clustering. We recognize the impossibility of providing a comprehensive survey of existing work on clustering and focus on revisiting the concepts most closely related to our context. Data-clustering

algorithms have a rich history since the early works such as [MacQueen et al. 1967], followed by significant research advancement, including the introduction of spectral clustering [Shi and Malik 2000; Von Luxburg 2007], center-based methods [Banerjee et al. 2005; Teboulle 2007], mixture models [Fraley and Raftery 2002], mean shift techniques [Comaniciu and Meer 2002], density-based spatial clustering (DBSCAN) [Ester et al. 1996], single-linkage technique [Gower and Ross 1969], affinity propagation algorithms [Frey and Dueck 2007], various adaptations of k-means clustering [Ahmed et al. 2020], and innovations in feature selection [Witten and Tibshirani 2010], among others. Recent literature on learning-based clustering, such as [Ahmed and Chew 2020; Chen et al. 2019], have focused on 3D point clouds, along with surveys [Ran et al. 2023; Ren et al. 2022].

Positioning wrt prior work. A popular proximity criterion that considers local density is the k-NN graph. However, it may not always be robust to noise in the data. On the other hand, the ϵ-neighborhood graph offers stability results and is known for its resilience to noisy datasets but does not take local density into account. Our new proximity formulation combines the advantages of both approaches by integrating the local density considerations of kNN with the robustness to noise inherent in the ϵ-neighborhood graph, also leading to a more comprehensive clustering solution, see Figures 2-3. As a complementary note, our approach shares similarities with diffusion distances used in spectral clustering, in the sense that we first re-embed the data with a new metric or dissimilarity, and then we use a standard clustering technique. What distinguishes our approach from spectral clustering is its particular simplicity, leading to an easy and scalable implementation. Notably, spectral clustering typically involves diagonalizing the matrix of the graph Laplacian operator, a process that does not scale well as dataset sizes increase. We also note that the simplicity of our proximity criterion distinguishes our clustering from ML-based approaches, which necessitate extensive training.

3 Method: proximity criterion & TDA

3.1 Context: Extending the SIG criterion

We begin by defining the sphere-of-influence graph, which is the foundation of our method. The SIG was first introduced as a proximity graph used for clustering [Toussaint 1988] but differs from widely used proximity graphs as it is not a subset of the Delaunay triangulation. In SIG, two vertices are connected by an edge if the distance between them is less than the sum of distances to their respective nearest neighbors. More formally, we connect a and b if

$$\mathrm{d}(a,b) \leq nn(a) + nn(b), \tag{1}$$

where $\mathrm{d}(a,b)$ is the distance between two points a and b and $nn(a)$ is the distance d between a and its nearest neighbor. This criterion exists in any dimension and can be modified to account for different types of data, such as disks, since the definition of the graph is purely based on distances.

This definition can also be viewed as a symmetrization of the nearest neighbor graph since the existence of an edge relies on the properties of both endpoints. While this definition manages to encode proximity well, as shown in [Marin et al. 2024], it cannot adapt to varying properties of the data, such as different densities or correlations. In this work, we consider an extension of this definition from two aspects: by considering generalized distance functions d, as well as by adding a parameter ε to Equation 1's inequality:

$$\mathrm{d}(a,b) \leq \varepsilon(nn(a) + nn(b)). \tag{2}$$

This allows us to change the connectivity of the proximity graph, capturing features with varying levels of significance. This parameter offers a novel degree of flexibility on top of the distance function d in defining the graph structure, allowing users to explore various cluster configurations. In Section 4.3, we also explore the possibility of employing local parameters. However, enabling users to select a parameter (even at the global level) for each dataset type poses challenges. Therefore, we offer an automatic parameter-tuning procedure based on tools from topological data analysis. As the creation of edges and their regrouping in our proximity graph relies on ε, we employ persistent homology to identify meaningful values for this parameter. We observe the formation of clusters (connected components in the SING graph) and analyze their duration. In the next sections, we first revisit key principles of Topological Data Analysis before presenting our contributions.

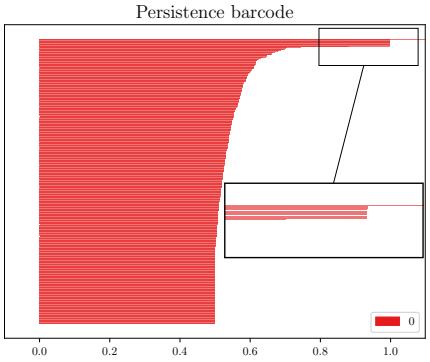


Fig. 4. Persistence Barcode. The zoomed-in region shows the stable region we are interested in. In a stable region, the number of connected components does not change in the specific interval, meaning that, most probably, those clusters carry some geometric meaning since ε needs to change a lot before collapsing connected components.

3.2 Background in TDA

In this section, we briefly review some of the material in TDA that will be used in the paper. For a more detailed introduction to the subject, we refer the reader to standard textbooks such as [Edelsbrunner and Harer 2010; Oudot 2015].

A *filtration* F of a topological space K over some totally ordered set T is a family $(\mathsf{F}_t)_{t\in T}$ of subspaces of K that are nested w.r.t. inclusion, that is: $\forall t \leq t' \in T$, $\mathsf{F}_t \subseteq \mathsf{F}_{t'}$. F is *simplicial* if K is a simplicial complex and if every F_t is a subcomplex of K.

The *(Vietoris-)Rips filtration* VR is a popular choice of simplicial filtration in TDA applications. Given a point cloud P equipped with a dissimilarity d, it is a filtration of the full simplex $K = 2^P$ (i.e., the power set of P viewed as a simplicial complex) indexed over $T = \mathbb{R}^+$, in which each simplex $\sigma = \{p_0, \cdots, p_m\} \subseteq P$ appears at index $t =$

$\max_{0\le i\le j\le m} \mathrm{d}(p_i, p_j)$. For any $t \in \mathbb{R}^+$, the subcomplex $\mathrm{VR}_t(P, \mathrm{d})$ of 2^P formed by those simplices that appear before or at t is called the *(Vietoris-)Rips complex* of P of parameter t.

$\mathrm{VR}(P, \mathrm{d})_t$ generalizes the t-ball graph of P in the following way: the vertices represent the points of P, and a simplex (not just an edge) exists if and only if its diameter is smaller than t. Varying the value of t from 0 to $+\infty$ gives the Rips filtration of (P, d).

When K is a finite simplicial complex (as is the case, e.g., for the Rips filtration and throughout this paper), applying homology in degree r with coefficients in some fixed field $\mathbf{k}$ to the filtration F yields a family of finite-dimensional $\mathbf{k}$-vector spaces connected by $\mathbf{k}$-linear maps. This family is called a *persistence module*. It is known to admit in this setting a complete algebraic invariant called the *persistence barcode* of F in degree r, denoted $\mathrm{B}_r(\mathsf{F})$, which takes the form of a finite multiset of intervals $[a_i, b_i)$, each of which encodes the lifespan of some topological feature of degree r appearing in the filtration. Note that multiple features can have identical lifespans, hence the multiset structure of the barcode. Topological features can be, for instance, connected components ($r = 0$), handles/tunnels ($r = 1$), enclosed voids ($r = 2$), or many other things. See Figure 4.

The *persistence diagram* of F in degree r, noted $\mathrm{PD}_r(\mathsf{F})$, is an alternative graphical representation of the barcode as a multiset of points above the diagonal $y = x$ in the plane. More precisely, every copy of the interval $[a_i, b_i)$ in $\mathrm{B}_r(\mathsf{F})$ becomes a copy of the point (a_i, b_i) in $\mathrm{PD}_r(\mathsf{F})$, and vice-versa. Throughout the paper, we will let $r = 0$, focusing on connected components in the filtration, and we will henceforth omit the parameter in our notations.

As multisets of points in the plane, persistence diagrams can be viewed as discrete measures in which each diagram point has unit mass. Such measures may have different total masses, therefore using classic distances between probability measures requires some adaptation. Typically, one enriches each diagram with infinitely many copies of the diagonal $y = x$ to even out the total masses, making no distinction between different infinite values. In this context, the *bottleneck distance* is the Wasserstein distance W^∞ between the enriched diagrams: $\mathrm{d_b}(\mathrm{PD}(\mathsf{F}), \mathrm{PD}(\mathsf{G})) := \mathrm{W}^\infty(\mathrm{PD}^+(\mathsf{F}), \mathrm{PD}^+(\mathsf{G}))$, where $\mathrm{PD}^+(\cdot)$ denotes the persistence diagram enriched with infinitely many copies of the diagonal $y = x$.

The persistence barcode or diagram of the Rips filtration exhibits the consistency (or lack thereof) of topological features hidden in the dataset across scales, thus it helps identify relevant scales at which to analyze or process the data. This methodology is backed by a sound stability theory, in particular by the fact that the map sending a point cloud P to the persistence diagram of its Rips filtration is provably Lipschitz continuous. In our proofs, we will use a more generic version of this stability theory, phrased as follows:

Theorem 3.1 (stability [Cohen-Steiner et al. 2007]). *Let K be a finite simplicial complex, and let $f, g\colon K \to \mathbb{R}$ assign a real value to each simplex in K. Then, the two families of sublevel-sets $f^{-1}((-\infty, t])$ and $g^{-1}((-\infty, t])$ for t ranging over $\mathbb{R}$ define two simplicial filtrations* F, G *of K such that:*

$$\mathrm{d_b}(\mathrm{PD}(\mathsf{F}),\, \mathrm{PD}(\mathsf{G})) \le \max_K |f - g|.$$

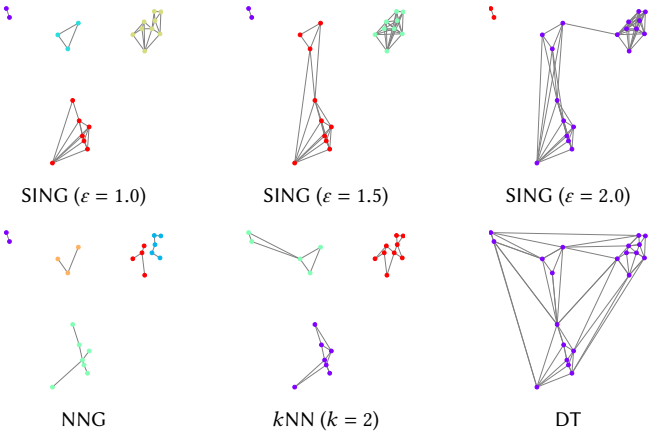


Fig. 5. As a proximity graph, SING encodes the evolution of various clusters without being limited by a predefined number of neighbors (as kNN) and offers more flexibility compared to Delaunay Triangulation (DT).

3.3 Key notion: density-sensitive semimetric

The proximity criterion introduced in Equation (2) can be interpreted as measuring the dissimilarity $\hat{\mathrm{d}}_P(a, b)$ between the points a, b against a threshold ε as follows:

$$\hat{\mathrm{d}}_P(a, b) := \frac{\mathrm{d}(a, b)}{nn(a) + nn(b)} \le \varepsilon. \tag{3}$$

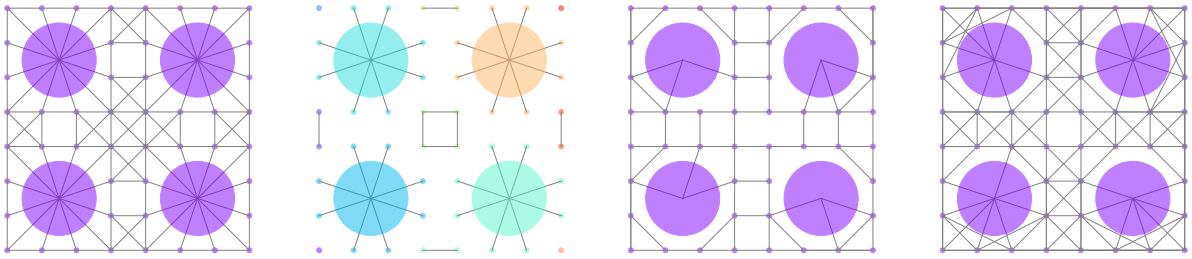
Observe that $\hat{\mathrm{d}}_P$ is a density-weighted version of the original metric d: consider indeed the 1-dimensional nearest-neighbor density estimator [Silverman 2018, §5.2], which is defined inversely proportional to the distance to the nearest data point. Dividing the distance $\mathrm{d}(a, b)$ by $nn(a) + nn(b)$ as in Equation (3) is equivalent, up to a constant factor, to multiplying $\mathrm{d}(a, b)$ by the harmonic mean of the 1-dimensional nearest-neighbor density estimates at a and b.

We also note that $\hat{\mathrm{d}}_P$ is defined only when P has at least 2 points, otherwise $nn(\cdot)$ itself is undefined. Note also that $\hat{\mathrm{d}}_P$ is a semimetric, not a metric, as it may not satisfy the triangle inequality. For instance, taking $P = \{-\eta, 0, 1/2, 1, 1+\eta\}$ on the real line gives $\hat{\mathrm{d}}_P(0, 1) = 1/2\eta$ and $\hat{\mathrm{d}}_P(0, 1/2) + \hat{\mathrm{d}}_P(1/2, 1) = 2/(1+2\eta)$, which infringes the triangle inequality as soon as $\eta < 1/2$ (and in fact makes the infringement as bad as it can be since $\hat{\mathrm{d}}_P(0, 1) \to +\infty$ while $\hat{\mathrm{d}}_P(0, 1/2) + \hat{\mathrm{d}}_P(1/2, 1) \to 2$ as $\eta \to 0$). However, the triangle inequality will not be needed in the following derivations.

3.4 SING and its connection to TDA

The Stability-Incorporated Neighborhood Graph (or SING for short) of (P, d) for parameter ε is defined as the ε-neighborhood graph of P in the semimetric $\hat{\mathrm{d}}_P$. By design, its connectivity adapts to the local density of the data. Moreover, it coincides with the 1-skeleton graph of $\mathrm{VR}_\varepsilon(P, \hat{\mathrm{d}}_P)$, so one can use the Rips filtration of $(P, \hat{\mathrm{d}}_P)$ and its persistence diagram to determine a suitable value for parameter ε.

Thus, the SING enjoys the same ease of use and flexibility as ε-neighborhood graphs in general while addressing their lack of sensitivity to the local density. Besides, as shown in Section 4.5 (Proposition 4.1), the Rips filtration associated with SING comes



(a) SING with a single connected component. Using radius information, we further extract the types of disks in the distribution.

(b) SING for $\varepsilon = 0.55$, where the main clusters are already forming - the main disks, along with grid elements.

(c) Nearest Neighbor Graph does not manage to capture enough information to fully encode the geometric features of the data.

(d) KNN for $k = 6$ does not manage to capture all the large disk connections, even if the grid pattern is almost fully represented.

Fig. 6. Structured pattern data connected using various proximity graphs.

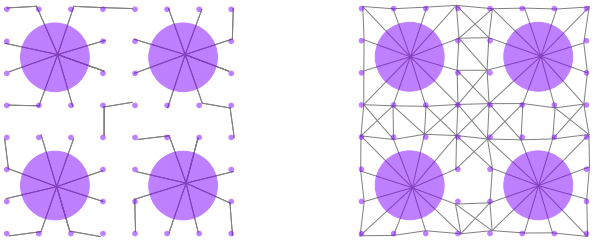
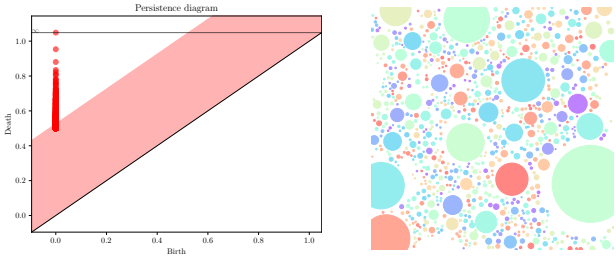


Fig. 7. Proximity graphs on pattern data perturbed with noise – SING still captures the original connectivity, showcasing the stability of the method.



(a) Confidence interval for given dataset.

(b) Example of noisy results for $\varepsilon = 0.5$, within the noise interval.

Fig. 8. Stability: extracting a confidence interval for our method using TDA. On the left, the red diagonal band spans the noisy values of our parameter, while on the right, we showcase clustering results for a noisy value. Small variations around this value result in multiple changes in terms of the number of clusters and no clear cluster structure is visible.

with theoretical stability guarantees, as does the Rips filtration in the ambient metric d, although in a weaker form.

Figure 3 illustrates the benefit of replacing the metric d by the weighted semimetric $\hat{\mathrm{d}}_P$ in the ε-neighborhood graph construction, in terms of the expressivity of the persistence barcode of its associated Rips filtration, missing the possibility of the four clusters.

We note that for some contexts, we might be interested in defining the SING complex accordingly as the $\mathrm{VR}_\varepsilon(P, \hat{\mathrm{d}}_P)$, whose 1-skeleton, as mentioned before, corresponds to our introduced SING graph. This is nicely illustrated in our example of Figure 2. Interestingly, the left-hand side figure shows both SING and the ε-neighborhood graph in the Euclidean distance, which coincide but, for different parameter values. However, the range of values that produce this "good" graph with SING is larger than its analog with the ε-neighborhood graph in the Euclidean distance—hence the interest in the SING. As a last remark, let us once again highlight the flexibility of SING in terms of the input metric. In a configuration similar to the one in Figure 2, if the desired clusters are indeed the ones presented on the right, incorporating an anisotropic metric favoring the diagonal direction would also enable us to achieve this clustering while maintaining stability.

4 SING Features and Advantages

We will now further analyze the graph and its properties, especially in comparison with other proximity graphs or other clustering techniques, highlighting the benefits of using such a stability-incorporated neighborhood graph.

4.1 Intrinsic Geometric Features

The graph naturally encodes proximity, and even with a global ε parameter for the entire dataset, the graph definition still inherently captures local density. Moreover, this graph is unlikely to suffer from long edges spanning the entire input or extremely high-degree nodes (except for adversarial cases – samples on a circle with the center of the circle). Unlike kNN graphs, we do not impose strict bounds on vertex degrees or distances between points. We compare our graph to popular proximity graphs in Figure 5.

SING can effectively capture both complex and regular geometric patterns, which are typically challenging to analyze and generate. Figure 6 illustrates such a configuration, which finds motivation in various contexts such as simulation. For example, in a spring system aiming to preserve a specific shape, our proximity graph seems more relevant to be employed for construction. Also, as expected,

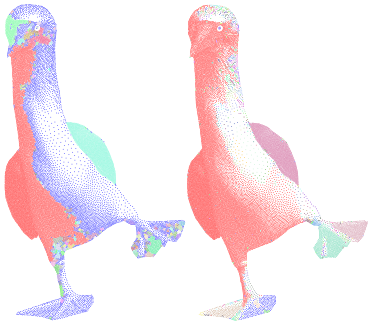


Fig. 9. Despite variations in density, our method (right) successfully extracts the bird's body as a single component, unlike DBSCAN (left), which separates it based on density fluctuations. Additionally, our method effectively distinguishes different parts of the feet into meaningful clusters. However, certain dense regions pose challenges for our method to cluster in a meaningful semantic way. Input point pattern from [De Goes et al. 2012].

in the presence of noise (as long as the extent is not too large), our graph still captures the original shape of the data, Figure 7.

4.2 Other distances

Since the definition of our graph only depends on distance computations, it can be used in any metric space and with various distance metrics. The persistence analysis benefits from the same advantages of only requiring distances between vertices for the computation. This flexibility allows us to compute the SING for other metrics, such as the disk-based distance of [Ecormier-Nocca et al. 2019], applied to disk distribution clustering. For two disks with radii r_1 and r_2, let d be the distance between the disks' centers, assuming, without loss of generality, that $r_1 \geq r_2$. Their disk distance is:

$$\mathrm{d}_{disks} = \begin{cases} f/(4r_1 - 4r_2) & d \leq r_1 - r_2 \\ (f - 4r_1 + 7r_2)/(3r_2) & r_1 - r_2 < d \leq r_1 + r_2 \\ f - 4r_1 + 2r_2 + 3 & \text{otherwise,} \end{cases} \tag{4}$$

$$f = \max(d + r_1 + r_2, 2r_1) \quad \text{(extent)} \tag{5}$$

$$- \operatorname{clip}(r_1 + r_2 - d, 0, 2r_2) \quad \text{(overlap)} \tag{6}$$

$$+ d + r_1 - r_2. \tag{7}$$

We feed this distance to our proximity criterion and conduct experiments for disk distribution clustering. The original data includes information about the class of each disk. However, in the absence of such information, the SING is perfectly able to extract similarities between different disks and output the relevant classification. This is done by analyzing the persistence barcode of the data and using the stable intervals as guidance for meaningful ε choices, leading to relevant clustering of data, Figure 1. As can be seen in the final result, the large disks are only connected to the smaller ones which they overlap, while the outer smaller disks are all connected in a single component, which exactly matches the input behavior. By using further details from the distance metric (i.e., classifying edges by the type of overlap they encode), we can easily extract the three classes present in the input. However, this dataset also showcases the limitations of a method that is only based on proximity – the single disk in the top-left corner, which is too far away from disks of similar class to be easily clustered with them.

4.3 Local density adaptation: flexible variant

Density constraints. In some point patterns, differentiating between different areas is not encoded by geometrically separating the points from different sections but by changing the density of the pattern (e.g., sparse points to represent the background of a stipple art image). To allow for different density encodings in our graph, we propose introducing a new parameter that enables edge creation only if the density is similar enough. We implement this by multiplying by the ratio between the nearest neighbors – Algorithm 1. We raise this ratio to a user-defined power, enforcing dissimilar densities to increase the measurement between points – Figure 13. Setting the density parameter to 0 brings us back to the original formulation.

ALGORITHM 1: Density SING computation.

Data: $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$, density ρ
Result: SING = $\{(a, b) : a, b \in P, a \neq b\}$

SING := {};
Find $nn(p) \forall p \in P$;
for *each pair (a,b)* | $a, b \in P, a \neq b$ **do**
  $\hat{\mathrm{d}}_P(a, b) := \|a - b\|_2 / (nn(a) + nn(b)) \times$
  $(\max(nn(a), nn(b))/\min(nn(a), nn(b)))^\rho$
end
PD := ComputePersistenceDiagram($\hat{\mathrm{d}}_P$);
ε := ExtractOptimalValue(PD) ;
for *each pair (a,b)* | $a, b \in P, a \neq b$ **do**
  if $\hat{\mathrm{d}}_P(a, b) \leq \varepsilon$ **then**
    SING := SING $\cup \{(a, b)\}$
  end
end

4.4 Complexity analysis

Nearest neighbor search complexity. Note that achieving sub-linear query time for nearest neighbor search in arbitrary dimension is a notoriously hard problem, especially in high dimensions, where concentration-of-measure phenomena occur. In low dimensions, there are classic data structures that allow for nearest neighbor search in sub-linear time. For instance, in 2D, one can build a hierarchical Delaunay triangulation in $O(n \log n)$ time and then use it for $O(\log n)$ time nearest neighbor queries. In 3D, this approach already has quadratic time complexity for the construction step in the worst case. Lastly, in arbitrary dimensions, locality-sensitive hashing allows achieving quasi-linear construction and then sub-linear query time.

SING Complexity. For $\varepsilon \leq 1$, the number of edges in SING is linear in the input size, resulting in an efficient algorithm in terms of space complexity. Regarding time complexity, the current $O(n^2)$ implementation of the SING algorithm, outlined in Algorithm 1, may be reduced to sub-quadratic construction time and even $O(n \log n)$ time in small dimensions (typically 2D), using classic data structures.

(a) Euclidean distance

(b) Our dissimilarity semimetric

Fig. 10. Dendrograms representing single linkage clustering of the dataset presented in Figure 2. The single linkage connects components with minimal distance, and the dendrogram illustrates the merging pattern. We observe that by using our semimetric, an additional large cluster (in blue) is formed. Thus, our semimetric captures cluster formations that would be missed by the Euclidean one.

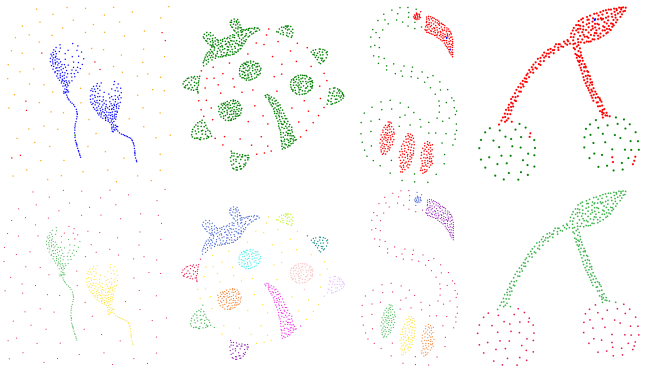


Fig. 11. Our clustering results (top) compared to DBSCAN (bottom). Our method connects similar densities in a single connected component due to the density-based proximity definition. The independent structures within the same density clusters can further be decomposed into distance-only connected components using a post-processing step.

4.5 Stability and Robustness

Let (X, d) be a metric space. Given a point cloud P in X, i.e., a finite subset $P \subseteq X$, we write $|P|$ for its cardinality and δ_P for its minimum pairwise distance: $\delta_P := \min_{p \neq q \in P} \mathrm{d}(p, q) > 0$. We equip the set of point clouds in X with the Wasserstein distance W^∞, which is valued in $\mathbb{R}^+ \cup \{+\infty\}$ and is finite whenever the two point clouds under consideration have the same cardinality:

$$\forall P, Q \subseteq X,\ \mathrm{W}^\infty(P, Q) := \begin{cases} \min\limits_{\substack{\gamma\colon P \to Q \\ \text{bijection}}} \max\limits_{p \in P} \mathrm{d}(p, \gamma(p)) \text{ if } |P| = |Q| \\ +\infty \text{ otherwise.} \end{cases} \tag{8}$$

This turns the set of point clouds into an extended metric space. Meanwhile, we equip the space of persistence diagrams with the bottleneck distance $\mathrm{d_b}$. Our stability guarantees are stated as follows:

Proposition 4.1. *The map $P \mapsto \mathrm{PD}(\mathrm{VR}(P, \hat{\mathrm{d}}_P))$ is continuous on the subspace of point clouds of cardinality at least 2 in X.*

The condition that $|P| \geq 2$ in the statement is not an artifact: it comes from the fact that $\hat{\mathrm{d}}_P$ is not defined when $|P| < 2$ – see Equation (3). The proof of Proposition 4.1 relies on the stability theorem for persistence diagrams (Theorem 3.1) and is provided in the supplemental material. Note that it requires the triangle inequality for the ambient metric d in X. However, in practice, as in the disk distance experiments, the proximity criterion is perfectly definable for semimetrics on X, leading to relevant graphs and desirable clusters – Figure 7.

Remark. The stability guarantee offered for SING by Proposition 4.1 is weaker than the one known for ε-neighborhood graphs in the ambient metric d [Chazal et al. 2014]. Primarily because it is only a continuity result, not a Lipschitz continuity result: our proof, although not tight, exhibits enough of the structure of $\hat{\mathrm{d}}_P$ to suggest that $\hat{\mathrm{d}}_P$ itself is not globally Lipschitz continuous but only locally Lipschitz continuous, with a local Lipschitz constant that grows with $1/\delta_P$ – see Equation 13 in the supplemental material. Secondarily, our stability guarantee is expressed in terms of the Wasserstein distance on point clouds in X, not in terms of the usual Hausdorff distance: in fact, there is no analog of Proposition 4.1 when the space of point clouds in X is equipped with the Hausdorff distance, as showed by the following counterexample. Given any positive $\varepsilon \leq 1/3$, consider two point clouds $P = \{0, 1\}$ and $Q = \{0, 1, 1 + \varepsilon\}$ on the real line $\mathbb{R}$. Their Hausdorff distance is ε. Meanwhile, we have $\hat{\mathrm{d}}_P(0, 1) = 1/2$ so $\mathrm{PD}(\mathrm{VR}(P, \hat{\mathrm{d}}_P)) = \{(0, +\infty);\ (0, 1/2)\}$, whereas $\hat{\mathrm{d}}_Q(0, 1) = 1/(1 + \varepsilon)$, $\hat{\mathrm{d}}_Q(1, 1 + \varepsilon) = 1/2$ and $\hat{\mathrm{d}}_Q(0, 1 + \varepsilon) = 1$ so $\mathrm{PD}(\mathrm{VR}(Q, \hat{\mathrm{d}}_Q)) = \{(0, +\infty);\ (0, 1/2);\ (0, 1/(1 + \varepsilon))\}$, hence $\mathrm{d_b}\left(\mathrm{PD}(\mathrm{VR}(P, \hat{\mathrm{d}}_P)),\ \mathrm{PD}(\mathrm{VR}(Q, \hat{\mathrm{d}}_Q))\right) = \frac{1-\varepsilon}{2(1+\varepsilon)}$. This quantity goes to $1/2$ while the Hausdorff distance between P and Q goes to zero as $\varepsilon \to 0^+$. Thus, the map $P \mapsto \mathrm{PD}(\mathrm{VR}(P, \hat{\mathrm{d}}_P))$ is not continuous in the Hausdorff distance.

5 Results and Validation

Parameters. In order to take advantage of the connection to TDA, we employ further analysis to extract stable intervals for our parameter. In persistence analysis, components that disappear shortly after creation are considered noise. Visually, the persistence diagram encodes this as points very close to the diagonal. We aim to extract

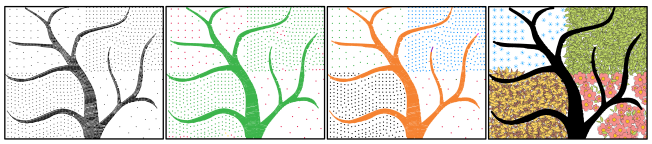


Fig. 12. Left to Right: Input stipples, Result of DBSCAN [Ester et al. 1996], Our clustering result, Result after replacing our clusters with stamps.

a stable interval for our parameter, where we are certain, up to a confidence ratio, that we will not encounter noisy components. We do this by subsampling our input set multiple times, with repetition, and computing the persistence diagram of each subset. We compute the distance from each of these subsampled diagrams to the original full input diagram and extract a band of possible noise around the diagonal (Figure 8). We then consider viable ε values the ones outside the computed noise band around the diagonal.

Performance & Implementation Details. All experiments were performed using an AMD Ryzen 7 5800 CPU. We implemented our method in `python`, using the Gudhi library for TDA and various packages from `scikit` and `sclearn` for data structures and the methods we compared to. The runtime of our current, non-optimized implementation spans from 1.5s for an input size of 500 points to 130s for 50k points. The source code is available online - https://github.com/dianam76/SING.

Applications. In addition to introducing our proximity criterion, we provide a category-based representation of its applications. Our analysis and validation of this method span various applications, including clustering and data segmentation, reconstruction (with a specific focus on 2D), stipple art coloring or editing, and network topology analysis, while briefly discussing potential advantages for anisotropic clustering, which falls outside the paper's scope.

5.1 Clustering and Data Segmentation

Clustering based on the SING-connected components integrates local density considerations into the ϵ-neighborhood graph and, consequently, into the Rips complexes, providing a significant generalization, as discussed for the example of Figure 2. Moreover, throughout our experimentation, we explored the most relevant clustering methods in terms of local density consideration, including k-means, density-based spatial clustering of applications with noise (DBSCAN), as well as the clustering induced by ϵ-neighborhood graphs. Among these, DBSCAN yielded the most optimal grouping outcomes in general, and we compare our results to their clusters in Figure 11. We analyze the evolution of our clusters compared to Single Linkage [Gower and Ross 1969] in Figure 10, showing how our proximity encoding captures more information about the input.

5.2 Multi-Class Disk Distribution Analysis

Distributions can also be represented through disk distribution, where the radius of the data can encode additional information. For example, the ecosystem examples presented in [Ecormier-Nocca et al. 2019] encode the size of the natural elements such as vegetation types. Such types of distributions are commonly used in artificial ecosystem generation in tools such as Ecobrush [Gain et al. 2017].

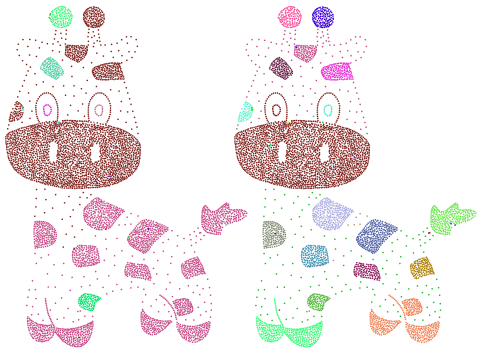


Fig. 13. On the left, clustering results without the incorporation of the density parameter, where samples are linked based on our proximity criterion. On the right, results incorporating the density parameter, effectively clustering the spots on the giraffe. Following a connected-component-based splitting, we can further refine the clustering.

For such disk distributions, the class of each disk has to be known in advance to be able to extract intra- and inter-class relationships. In Figure 1, we are able to extract the classes (as individual clusters) given only the input coordinates and radii. We compute the persistence barcode of our data by varying the ε parameter (as in Figure 4). We then use the stable intervals in the diagram to guide the parameter choice in the direction of the most meaningful clusters (Figure 1, right). We could further group all large disks in the same class using the topology of our SING graph (in this case, by observing they all have the same type of connections), considering filters on the disk distance if needed (in this case, the distance values reflect the fact that all of their neighbors are placed inside the disk).

5.3 Stipple Art Manipulation

Stipples are patterns of points where the visual information is encoded through density and correlation. Clustering stippling patterns into visually meaningful regions is challenging and does not necessarily align with our visual perception. Despite this lack of ground truth, SING clustering provides promising results, as showcased by Figure 15. For challenging density-varying stipples, we use the SING variation that accounts for the density parameter. Some examples are shown in Figure 11, where our results for layer extraction are similar to DBSCAN [Ester et al. 1996]. However, selecting parameters for DBSCAN relies only on data properties, lacking an easily inferred optimal parameter value, in contrast to our method. Note

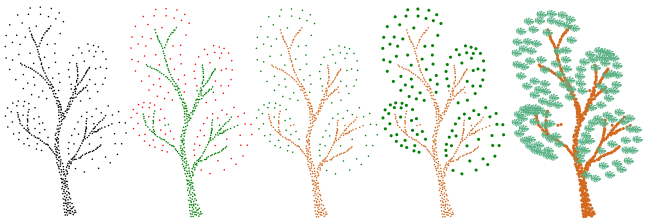


Fig. 14. Left to Right: Input stipples, clustering result, the result after varying color, the result after editing stipple size, replacing stipple with pattern.

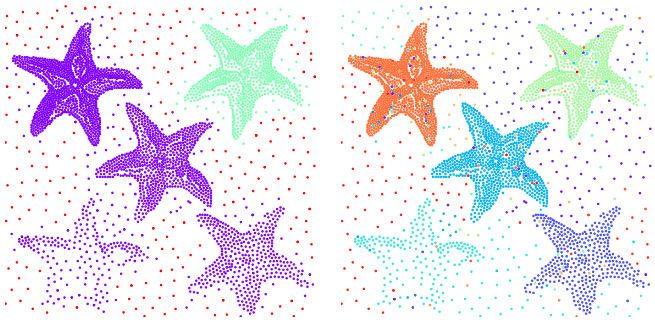


Fig. 15. A point pattern synthesized from an image, using the method in [Huang et al. 2023], is segmented using the DBSCAN and SING clustering.

(a) Boundary extraction on stipple data. (b) Boundary extraction after our clustering. (c) Boundary extraction after DBSCAN clustering.

Fig. 16. Boundary extraction of stipple art using Discern [Thayyil et al. 2021]. All boundary methods we have tested fail on inputs with multiple densities. However, running them on clustered input results in meaningful boundaries.

(a) Original road network G. (b) Rips complex restricted to G. (c) SING filtration on the input graph.

Fig. 17. Network simplification under a fixed edge budget, utilizing the Oldenburg road network [Mokbel et al. 2004]. Our results enable improved overall connectivity in the generated network graph (decreasing the connected components count – 1751 compared to 1839).

that connecting samples by local density may form small, packed clusters, as seen in the cherry example. This can be adjusted via the ϵ parameter or interpreted as a feature in post-processing, given the absence of a ground-truth segmentation in stipple art. Moreover, in Figure 12, the slight variation in density is not captured by DBSCAN. Having such a layered representation allows for easy and meaningful manipulation of the art, like editing the distribution [Huang et al. 2023] and representation (see Figure 14). Automatically generated stipple patterns that exhibit varying density require more parameter tuning, as the difference between distinct portions of the input image is not sharp, and changing density across an area is used to illustrate various visual effects – Figure 9.

5.4 2D Reconstruction

Shape reconstruction is the process of identifying the shape induced by a set of points [Methirumangalath et al. 2015; Thayyil et al. 2020, 2021]. This is a well-known ill-posed problem in Computational Geometry, with applications in GIS. Thanks to SING, as shown in Figure 16, we can easily group points in a meaningful manner and then extract their boundary to determine the shape. Additionally, with our SING variant, we can further extend the traditional shape reconstruction problem by considering density variations and approaching a level of efficiency closer to human perception in this context. Extracting the boundary directly from our graph is promising as well and worth future investigation.

5.5 Network Topology Analysis

Network graph analysis and classification involves understanding the overall structure of the graph and making predictions based on that structure [Kartun-Giles and Bianconi 2019]. In our geometric context, we experiment on spatial networks, which have a clear positional embedding. Leveraging our persistence-based proximity criterion, which effectively captures topological characteristics across different scales, enables us to gain a comprehensive understanding of the graph's shape and connectivity. As a result, our approach encodes, under the same edge budget, a more meaningful simplification of original data compared to the Rips complex, capturing the original shape of the road network better - Figure 17.

6 Discussion and Perspectives

Limitations. Figure 9 shows various clustering imperfections produced by our method. This is due to variations in the density information and the fact that the current version of our method does not incorporate explicit part labels or "semantic" knowledge. It also motivates a semantic extension of our work in order to disambiguate such cases. Additionally, while the selection of a suitable value for ε is largely guided by TDA, a comparable approach for determining the density parameter is currently lacking in our current implementation. For further efficiency improvement, optimizing our bottleneck distance computation could also lead to better runtime.

Future work. Subsampling and point set simplification both seem to be natural contexts in which our proximity criterion can be formalized and employed. Surface curvature and anisotropic metrics incorporation were already mentioned as promising inspirations for future work. Another theoretical future direction would be to investigate whether the SING is a spanner and to characterize the corresponding stretch factor. This is based on the fact that for ε tending to infinity, SING approaches the complete graph – a 1-spanner from a threshold ε value. Since our similarity metric extends straightforwardly to higher dimensions with low computational cost, it would also allow for the analysis of group behavior in animal swarms based on object-detection input, for example. Furthermore, the improved

connectivity can assist in creating richer features in images or point data, such as for photogrammetry or training networks on point data. This extension to higher dimensions and related application scenarios are left for future work due to our current unoptimized implementation and its being beyond this paper's scope.

Acknowledgments

This work has been partially funded by the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) project ICT19-009 and by the ANR JCJC project SketchMAD (ANR-23-CE33-0009). We acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme. We also extend our thanks to Pierre Ecormier-Nocca for providing the ecosystem dataset. We wish to thank Xingchang Huang for providing the data of Figure 15.

References

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. 2020. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* 9, 8 (2020), 1295.

Syeda Mariam Ahmed and Chee Meng Chew. 2020. Density-based clustering for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10608–10617.

David Avis and Joe Horton. 1985. Remarks on the sphere of influence graph. *Annals of the New York Academy of Sciences* 440, 1 (1985), 323–327.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. 2005. Clustering with Bregman divergences. *Journal of machine learning research* 6, 10 (2005).

Bentley and Ottmann. 1979. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on computers* 100, 9 (1979), 643–647.

Frédéric Chazal, Vin De Silva, and Steve Oudot. 2014. Persistence stability for geometric complexes. *Geometriae Dedicata* 173, 1 (2014), 193–214.

Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. 2019. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4994–5002.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. 2007. Stability of Persistence Diagrams. *Discrete Comput. Geom.* 37, 1 (Jan. 2007), 103–120. https://doi.org/10.1007/s00454-006-1276-5

Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24, 5 (2002), 603–619.

Luiz Henrique de Figueiredo and Afonso Paiva. 2022. Region reconstruction with the sphere-of-influence diagram. *Computers & Graphics* 107 (2022), 252–263.

Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. 2012. Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–11.

Pierre Ecormier-Nocca, Pooran Memari, James Gain, and Marie-Paule Cani. 2019. Accurate Synthesis of Multi-Class Disk Distributions. *Computer Graphics Forum* 38, 2 (2019), 157–168. https://doi.org/10.1111/cgf.13627

Herbert Edelsbrunner and John L Harer. 2010. *Computational topology: an introduction*. American Mathematical Society.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. Density-based spatial clustering of applications with noise. In *Int. Conf. knowledge discovery and data mining*, Vol. 240.

Chris Fraley and Adrian E Raftery. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association* 97, 458 (2002), 611–631.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

J. Gain, H. Long, G. Cordonnier, and M.-P. Cani. 2017. EcoBrush: Interactive Control of Visually Consistent Large-Scale Ecosystems. *Computer Graphics Forum* 36, 2 (2017), 63–73. https://doi.org/10.1111/cgf.13107 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13107

John C Gower and Gavin JS Ross. 1969. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 18, 1 (1969), 54–64.

L Guibas, J Pach, and M Sharir. 1992. Generalized sphere-of-influence graphs in higher dimensions. *Manuscript, Tel-Aviv University* (1992).

Xingchang Huang, Tobias Ritschel, Hans-Peter Seidel, Pooran Memari, and Gurprit Singh. 2023. Patternshop: Editing Point Patterns by Image Manipulation. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.

Jerzy W Jaromczyk and Godfried T Toussaint. 1992. Relative neighborhood graphs and their relatives. *Proc. IEEE* 80, 9 (1992), 1502–1517.

Alexander P. Kartun-Giles and Ginestra Bianconi. 2019. Beyond the clustering coefficient: A topological analysis of node neighbourhoods in complex networks. *Chaos, Solitons & Fractals: X* 1 (2019), 100004. https://doi.org/10.1016/j.csfx.2019.100004

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

Diana Marin, Stefan Ohrhallinger, and Michael Wimmer. 2022. SIGDT: 2D curve reconstruction. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 25–36.

Diana Marin, Stefan Ohrhallinger, and Michael Wimmer. 2024. Parameter-Free Connectivity for Point Clouds. In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - GRAPP*. INSTICC, SciTePress, 92–102. https://doi.org/10.5220/0012394900003660

Subhasree Methirumangalath, Amal Dev Parakkat, and Ramanathan Muthuganapathy. 2015. A unified approach towards reconstruction of a planar point set. *Computers & Graphics* 51 (2015), 90–97. https://doi.org/10.1016/j.cag.2015.05.025 International Conference Shape Modeling International.

Mohamed Mokbel, Xiaopeng Xiong, and Walid Aref. 2004. SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (04 2004). https://doi.org/10.1145/1007568.1007638

Steve Y Oudot. 2015. Persistence theory: from quiver representations to data analysis. *Mathematical Surveys and Monographs* 209 (2015), 218.

Xingcheng Ran, Yue Xi, Yonggang Lu, Xiangwen Wang, and Zhenyu Lu. 2023. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review* 56, 8 (2023), 8219–8264.

Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. 2022. Deep clustering: A comprehensive survey. *arXiv preprint arXiv:2210.04142* (2022).

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22, 8 (2000), 888–905.

Bernard W Silverman. 2018. *Density estimation for statistics and data analysis*. Chapman & Hall.

Marc Teboulle. 2007. A Unified Continuous Optimization Framework for Center-Based Clustering Methods. *Journal of Machine Learning Research* 8, 1 (2007).

Safeer Babu Thayyil, Amal Dev Parakkat, and Ramanathan Muthuganapathy. 2020. An input-independent single pass algorithm for reconstruction from dot patterns and boundary samples. *Computer Aided Geometric Design* 80 (2020), 101879. https://doi.org/10.1016/j.cagd.2020.101879

Safeer Babu Thayyil, Jiju Peethambaran, and Ramanathan Muthuganapathy. 2021. A sampling type discernment approach towards reconstruction of a point set in R2. *Computer Aided Geometric Design* 84 (2021), 101953. https://doi.org/10.1016/j.cagd.2020.101953

Godfried T Toussaint. 1988. A graph-theoretical primal sketch. In *Machine Intelligence and Pattern Recognition*. Vol. 6. Elsevier, 229–260.

Remco C Veltkamp. 1992. The γ-neighborhood graph. *Computational Geometry* 1, 4 (1992), 227–246.

Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (2007), 395–416.

Daniela M Witten and Robert Tibshirani. 2010. A framework for feature selection in clustering. *J. Amer. Statist. Assoc.* 105, 490 (2010), 713–726.