



HAL
open science

Incremental Delaunay triangulation construction for clustering

Octavio Razafindramanana, Frédéric Rayar, Gilles Venturini

► **To cite this version:**

Octavio Razafindramanana, Frédéric Rayar, Gilles Venturini. Incremental Delaunay triangulation construction for clustering. 22nd International Conference on Pattern Recognition, Aug 2014, Stockholm, Sweden. hal-01094436

HAL Id: hal-01094436

<https://hal.science/hal-01094436>

Submitted on 2 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Incremental Delaunay triangulation construction for clustering

Octavio Razafindramanana, Frédéric Rayar, Gilles Venturini
Université François Rabelais de Tours - Laboratoire d'Informatique
64, avenue Jean Portalis - 37200 Tours, France
{last}@univ-tours.fr

Abstract—In this paper, we propose an original solution to the problem of point cloud clustering. The proposed technique is based on a d -dimensional formulated Delaunay Triangulation (DT) construction algorithm and adapts it to the problem of cluster detection. The introduced algorithm allows this detection as along with the DT construction. Precisely, a criterion that detects the occurrences of gaps in the simplices perimeter distribution within the triangulation is added during incremental DT construction. This allows then to detect and label simplices as inter- or intra cluster. Experimental results on 2D shape datasets are presented and discussed in terms of cluster detection and topological relationship preservation.

Keywords—Computational geometry, Delaunay triangulation, clustering, forward approach

I. INTRODUCTION AND RELATED WORK

“One important goal of data analysis is to allow the user to obtain knowledge about the data” [1]. Referring to Carlsson’s discussions [1], one very basic objective of data analysis is to construct *qualitative geometric information* out of *quantitative information* by means of analyzing pairwise distances among the data points in order to reveal connected components. The visualization of this so-called *qualitative information* gives rise to the need for suitable tools to embed the *shape* of the data points, or subsets of points, and their *topological relationship* with respect to each others. Geometrical configuration preservation, or *topology preservation*, has then to be met for the sake of detecting denser parts of the point cloud (*clusters*) by means of visualizing a graph.

Proximity graphs [2] might satisfy this fundamental requirement as they might faithfully embed both local (neighborhood) and *high-level information* (groups of points) that exist among the data points. The *topology preserving* property of these graphs relies on how one might define proximity and this proximity defines how they would faithfully or unfaithfully encapsulate and summarize the distribution of the pairwise distances. Notable proximity graphs include Delaunay’s [3], relative neighborhood’s (RNG) [4], Gabriel’s [5], minimum spanning tree (MST) [6], k -nearest neighbor (k -NN) or β -skeleton’s [7].

Furthermore, referred to as graph theory-based techniques in the clustering literature [8], [9], usage of proximity graphs for the task of clusters detection is notably made in algorithms such as CHAMELEON (k -NN) [10], Zahn’s (MST) [11], Mulhenbach’s (any proximity graph) [12], MOSAIC (RNG) [13], AUTOCLUST and TRICLUST (Delaunay) [14], [15].

However, none of these algorithms take any *volumetric information* into account as they all focus on analyzing the edge lengths distribution. In particular, none of the cited Delaunay based algorithms take into account the amount of topological information carried by the simplices. The works presented in [16] and [17] address this problem as the former takes into account the distribution of the perimeters of the simplices and the latter proposes a composite measure based on the association of the perimeter, intra-simplex edge length distribution and local information. These two algorithms therefore refine the resulting graph by making use of the information within the *simplicial complex* [18]: in opposition to graph theory-based techniques, they are *simplicial complex-based* ones.

Moreover, proximity graph-based clustering algorithms rely on: 1) the construction of the proximity graph, 2) the revelation of the possibly existing clusters by means of pruning extra edges. In this paper, we then argue about the possibility of detection of clusters among the data points as along with the proximity graph construction process. To this end, we introduce a one-pass *forward strategy*. In particular, we propose to adapt a d -dimensional Delaunay triangulation (DT) algorithm, namely *DeWall* algorithm [19], for the task of clustering. Precisely, the proposed criterion detects breaks of homogeneity in the perimeter distribution as along with the insertion of a new simplex. The detection of these breaks of continuity in the distribution allows then to label the newly inserted simplex as being *inter* or *intra-cluster*. The underlying problem addressed throughout this paper is therefore revealing the possibly existing clusters among a set of points as along with the construction of their DT.

The presented work is defined for a finite data points set $S \subset \mathbb{R}^d$ and outputs the Delaunay triangulation of a as along with an unoriented graph $G = (V, E)$ such that $V = S$ and $E \subseteq S \times S$ are its vertices and edges sets respectively.

II. PROXIMITY GRAPHS AND DELAUNAY TRIANGULATION

A. Proximity graphs and clustering

Let us define the general criterion for the creation of a given edge e_i that belongs to proximity graph G :

$$e_i \in E \Leftrightarrow I(P) \cap \{V \setminus P\} = \emptyset, \forall i = 1 \dots C_k^2 \quad (1)$$

with $P = \{p_1 \dots p_k\} \subset V$ a given set of affinely independent points ($2 \leq k \leq |V|$) and $I(P)$ a *conflict zone* in which existence of points is tested. The edges of E , that state that given data points $P \subseteq S$ are *close* to each others, are such

that the zone $I(P)$ contain no other points than the one of P . This zone might for instance be defined as a d -lune or d -ball in case of *RNG* ([4]) or *Gabriel* ([5]) graphs respectively.

In the presented work, E is populated by edges satisfying the *empty ball* Delaunay criterion [3], i.e. $P = \{p_0 \dots p_d\}$ and $I(P)$ is the hypersphere passing through the $d+1$ points of P . Furthermore, dataset $V = S$ is assumed to be in *general position*, i.e. there is no $d+2$ points of S lying on a common hypersphere. In that non-degenerate configuration, the DT of S is then uniquely defined (and therefore G as well). From now on, we shall denote $DG(S)$ the Delaunay graph of S and $DT(S)$ the Delaunay triangulation of S .

B. Techniques for constructing the Delaunay triangulation

1) *Definitions and elementary properties*: The Delaunay complex or Delaunay triangulation (DT) is a widely-used structure that allows to grasp geometrical and topological relationships between given d -dimensional points. The graph $DG(S)$, composed of all the edges of $DT(S)$, present the well-known following property that

$$MST(S) \subseteq RNG(S) \subseteq GG(S) \subseteq DG(S)$$

as a result of embedding the edges of a *simplicial complex* (i.e. the set of all the k -simplices, $k \in [0, d+1]$). As well, each edge of $DG(S)$ denotes the adjacency of two *Voronoi cells* within the *Voronoi diagram* of S [18].

A point $x \in \mathbb{R}^d$ is a vertex of $DT(S)$ iff $x \in S$ and $DT(S)$ has several elementary properties among which we enumerate the following two:

- 1) The intersection of two simplices in $DT(S)$ is either an empty set or a common face.
- 2) The circumscribed hyper-sphere passing through the $d+1$ vertices of each simplex σ does not contain any other point of S in its interior (empty *conflict zone* mentioned in equation (1)).

Furthermore, each d -simplex is defined by the convex hull of $d+1$ points of S (e.g. triangles or tetrahedron when $d=2$ or $d=3$ resp.) and defines $d+1$ ($d-1$)-simplices (e.g. 3 faces in dimension 2). Formally, each d -simplex is composed of k -simplices (composed of $k+1$ points), with $0 \leq k \leq d$. Finally, a *simplicial complex* is a finite collection of faces C such that:

- 1) Any face of C is a k -simplex ($0 \leq k \leq d$).
- 2) Any sub-face of any face of C belongs to C .
- 3) The intersection of any face of C is either empty or a face of C .

According to [20], sequential algorithms for constructing the Delaunay triangulation are composed of the following five types of algorithms: *sweep-line*, *lift method based*, *incremental*, *divide-and-conquer* and *gift-wrapping*. As the presented method is based on the last three paradigms, let us now briefly describe these three types of algorithms. Exhaustive insight about Delaunay triangulation algorithms might be found in [18].

In a nutshell, these algorithms incrementally insert a new point $p \in S$ and are divided in two types regarding 1) if

they incrementally add new simplices outside the convex hull of the current triangulation or 2) if they *add new simplices* unpredictably inside or outside the convex hull of the current triangulation. This boils-down to knowing if *to-be-inserted point* p would lie outside current convex hull or unpredictably inside or outside current convex hull. On the one hand, in the latter case [21], [22], the incremental insertion of a new point p implies the detection of the *conflicting* simplices i.e. simplices whose circumscribe hypersphere contain p . The current triangulation is to be modified quite surely and finding the conflicting simplices is a making bottleneck. On the other hand, algorithms such as [23] do not modify the current triangulation and only *wraps* it with new simplices, therefore inserting point p outside the convex hull of current unfinished triangulation. In addition, during the *gift-wrapping* process, a dictionary of *unfinished* faces that are on the boundary of the current triangulation is maintained, as a result of adding simplices outside the current convex hull. These *unfinished* faces are still up to be processed and the *gift-wrapping* process stops when there is no *unfinished d-face* anymore.

Finally, *divide-and-conquer* algorithms split the points set S in two disjoint subsets S_1 and S_2 . Then, the triangulation is recursively computed on S_1 and S_2 . In the end, partial results, covering S_1 and S_2 , are merged to obtain actual $DT(S)$. The merging of the partial triangulations constitute the bottleneck of these techniques and attention has to be paid to the splitting hyperplanes choice strategy.

C. The DeWall algorithm : a divide-and-conquer space-filling strategy

The strategy we based our method on, namely DeWall algorithm [24], is an algorithm mixing both *gift-wrapping* and *divide-and-conquer* DT construction paradigms : starting from an initial Delaunay d -simplex crossed by an axis-aligned hyperplane α , the triangulation is constructed by constructing α -crossed simplices, incrementally adding points lying in the outer part of current convex hull (*gift-wrapping*) until no further point satisfying this condition can be found. For a given α , following this rationale then leads to a partition of V into three parts : the one composed of vertices belonging to the d -simplices crossed by α (the *wall*) and the two other ones lying on each side of the *wall*. The algorithm is then recursively applied on these two parts (*divide-and-conquer*) and thus implicitly 1) creates a sequence of axis-aligned hyperplanes in a fashion kd -trees do, 2) constructs triangulation *wrapping* a space-filling structure. An overview of the execution of the algorithm is illustrated in Figure 1.

The actual incremental construction proceeds as follows: given a face f of simplex σ , the unique simplex that shares face f with σ is built. This construction implies finding the point $p^* \in S$ which satisfies the empty hypersphere property. As formulated in [24], p^* is the point which informally minimizes the radius of circumscribed hypersphere and formally minimizes the *Delaunay distance* (dd) to f :

$$dd(f, p) = \begin{cases} r : c \in \text{Half}(f, p) \\ -r : \text{otherwise} \end{cases}$$

with r and c the radius and center of the circumsphere passing by p and extremity points of face f . $\text{Half}(f, p)$ is the half-

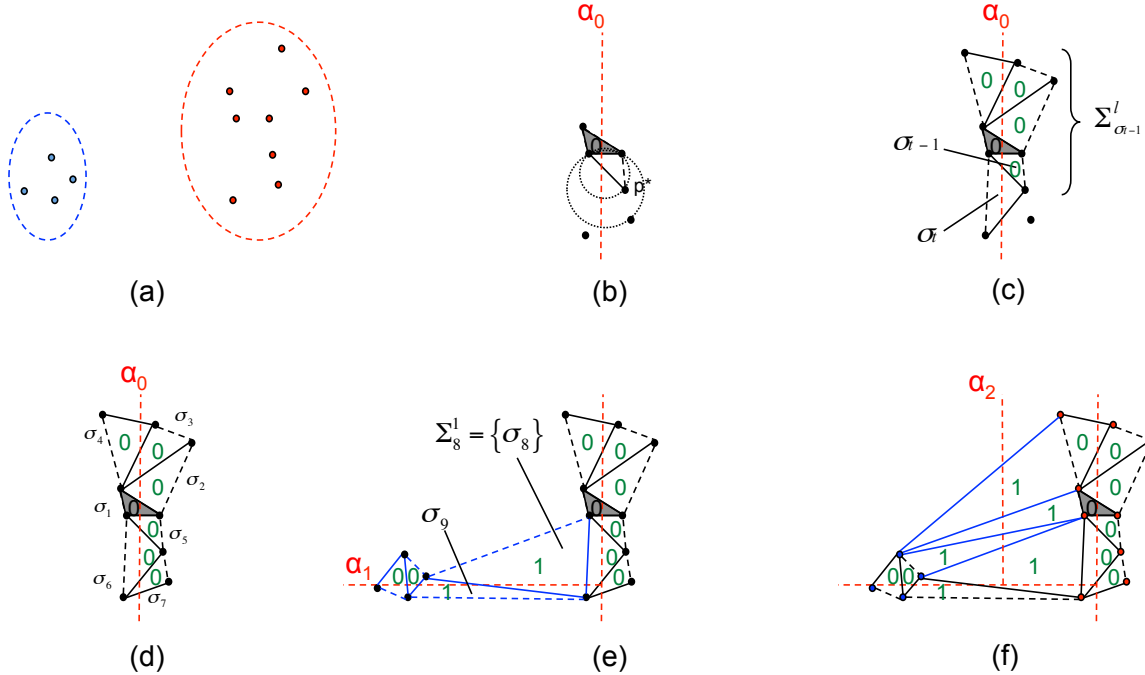


Fig. 1. **The proposed one-pass strategy in dimension 2.** Plain blue edges are the current wall's finished ones. Successive hyperplanes are drawn in dashed red. The black dotted edges are the remaining unfinished faces of the triangulation. **(a) Toy dataset** 12 points divided in two separated clusters: the blue and the red one; **(b) First simplex and hyperplane** Hyperplane α_0 , and initial α_0 -crossed Delaunay simplex (shaded) are selected. p^* is then computed such as it minimizes the radius of the circumscribed Delaunay hypersphere passing through current unfinished face; **(c) Label propagation** Simplex σ_t is added to the ones of $\Sigma_{\sigma_{t-1}}^l$ and the intra-cluster label $l = 0$ is propagated. At current iteration t , σ_t is τ times larger than σ_{t-1} thus σ_t may be wrongly labeled as an inter-cluster if not compared to $\Sigma_{\sigma_t}^l$; **(d) First wall construction** Simplices are added until no point lies outside the unfinished α_0 -crossed faces: first wall construction ends here; **(e) Second wall construction** σ_8 is labeled as inter-cluster as it is τ times as large as the average one of $\Sigma_{\sigma_6}^0$. σ_9 is not τ times as small as σ_8 and therefore shares the same label as σ_8 . Note that if σ_9 would have been τ times as small as σ_8 , as $|\Sigma_{\sigma_8}^1|$ equals 1, σ_9 would have been labeled as intra-cluster as it presents a perimeter smaller than the minimum one of $\Sigma_{\sigma_9}^{inter}$; **(f) Cluster detection.** The algorithm continues until no *unfinished* faces remain. At this stage of construction, the algorithm correctly detected two connected components that correspond to the blue and red cluster of the toy dataset.

space defined by the hyperplane passing through f and by the point p .

III. ABOUT THE INSERTION OF A CLUSTER DETECTION CRITERION IN THE TRIANGULATION CONSTRUCTION

A. DT-based clustering algorithms

Along with the proximity graphs based ones, the existing DT-based clustering techniques leverage the following paradigm: (i) $DT(S)$ is computed, (ii) statistical information is extracted on the basis of the pairwise distances distribution analysis, (iii) the final graph is produced by pruning extra edges (or simplices) with respect to the previous statistical information. This *three-pass* procedure eventually leads to revealing the possibly existing clusters. Notable clustering algorithms following this paradigm are AUTOCLUST [14] and TRICLUST [15]. Other works ([16], [17]) do not only take the pairwise distances information into account but also the inherent volumetric information carried by the d -simplices. We call *forward-backward approaches* these *three-pass* algorithms.

As opposed to these techniques, we propose in this paper a *forward approach* as the clusters are detected along

with the construction of $DT(S)$. This approach leverages the incremental DeWall algorithm presented in Section II.C. Straightforwardly, DeWall algorithm allows to add a gap detection criterion at each insertion of a new simplex. As the triangulation is final after each insertion (*i.e.* previous simplices will not be modified in the next stages), it is possible to maintain statistical information of the current stage of the construction of $DT(S)$ and therefore to propagate the *inter-cluster label* accordingly. The proposed inter-cluster labeling strategy is detailed hereunder.

B. A forward approach for cluster detection

The objective of the presented approach boils down to the selection of a suitable measure for the labeling of inter-cluster simplices at any stage of the construction of the triangulation. Derived from discussions in [25], [17], the proposed technique detects the occurrence of breaks of continuity in the distribution of the simplex perimeters. Indeed, following a density-based definition of clusters, clusters are groups of points among which distances are *significantly* smaller than the ones between the clusters: *void* surrounds the denser parts that constitute the clusters. Consequently, inter-cluster simplices are

prone to present higher perimeter values than the intra-cluster ones. We note $\delta_l(\sigma)$ the label of a given simplex σ , where $\delta_l(\sigma) = 1$ if σ is labeled as inter-cluster and 0 otherwise.

Let σ_t be the newly inserted simplex sharing a face with a previous simplex σ_{t-1} , $\Sigma_{\sigma_t} = \{\sigma_1, \dots, \sigma_t\}$ be the set of all the simplices previously constructed up to σ_t , $\Sigma_{\sigma_t}^{intra} = \{\sigma \in \Sigma_{\sigma_t}, \delta_l(\sigma) = 0\}$ be the set of intra-cluster simplices of Σ_{σ_t} , $\Sigma_{\sigma_t}^{inter} = \{\sigma \in \Sigma_{\sigma_t}, \delta_l(\sigma) = 1\}$ be the set of inter-cluster simplices of Σ_{σ_t} and finally

$$\Sigma_{\sigma_t}^l = \{\sigma_{t-k+1}, \dots, \sigma_t\}$$

such as $\delta_l(\sigma_{t-k}) = \dots = \delta_l(\sigma_t)$ and $\delta_l(\sigma_{t-k}) \neq \delta_l(\sigma_t)$.

$\Sigma_{\sigma_t}^l$ is the set of the previous simplices that share the same label as σ_t . Then, for a given iteration $t > 1$ and a given scalar $\tau > 0$, let $\delta(t, \tau) : \mathbb{N} \times \mathbb{R} \rightarrow \{0, 1\}$ be the inter-cluster labeling function such as

if $\delta_l(\sigma_{t-1}) = 0$,

$$\delta(t, \tau) = \begin{cases} 1 : \frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)} > \tau \\ 0 : \text{otherwise} \end{cases} \quad (2)$$

if $\delta_l(\sigma_{t-1}) = 1$,

$$\delta(t, \tau) = \begin{cases} 0 : \frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)} < \frac{1}{\tau} \\ 1 : \text{otherwise} \end{cases} \quad (3)$$

where $\rho(\sigma) = \sum_{e \in \sigma} |e|$ denotes the perimeter of the simplex σ , $|e|$ the length of e and $\text{mean}(S)$ (respectively $\text{min}(S)$ and $\text{max}(S)$) the average (respectively the minimum and maximum) perimeter value of the simplices in S .

As mentioned in equations (2) and (3), the labeling strategy of a newly inserted simplex is different regarding whether previously inserted simplex is labeled as intra- or inter-cluster. Indeed, if previous simplices of $\Sigma_{\sigma_{t-1}}^l$ were labeled as intra-cluster, the only configuration for which σ_t would not be labeled as intra-cluster would be that its perimeter is τ times as large as the average one of $\Sigma_{\sigma_{t-1}}^l$. Conversely, if the simplices of $\Sigma_{\sigma_{t-1}}^l$ were labeled as inter-cluster, the only configuration for which the label of σ_t would not be inter-cluster would be that the average perimeter of $\Sigma_{\sigma_{t-1}}^l$ is τ times as large as σ_t 's one, i.e. $\frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)} < \frac{1}{\tau}$. In other words, the label of $\Sigma_{\sigma_{t-1}}^l$ is propagated to its newly inserted neighbor σ_t unless σ_t presents a significant difference in size. In particular, a gap is detected between σ_t and σ_{t-1} if σ_t is τ times as large (small) as the average simplex of $\Sigma_{\sigma_{t-1}}^l$. Then, the computation of ratio $\frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)}$ allows to detect the so-called breaks of continuity in the perimeter distribution, and therefore the label changes, in agreement with the simplices of $\Sigma_{\sigma_{t-1}}^l$. Moreover, it provides a robust solution to the problem of mislabeling intra-cluster simplices as inter- ones as the perimeter distribution of the intra-cluster simplices might not to be locally homogeneous. The converse reasoning is followed to avoid labeling locally smaller inter-cluster simplices as intra-cluster ones (Figure 1).

Nevertheless, at each label change, $|\Sigma_{\sigma_t}^l|$ equals 1, therefore computing $\frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)}$ boils down to comparing σ_t to its neighbor, namely computing $\frac{\rho(\sigma_t)}{\rho(\sigma_{t-1})}$. This situation is prone to detect gaps that do not define the border of the clusters. In

that specific case, σ_t is compared to all the simplices computed since the beginning of the triangulation.

Thus, $|\Sigma_{\sigma_t}^l| = 1$ implies that:

if $\delta_l(\sigma_{t-1}) = 0$,

$$\delta(t, \tau) = \begin{cases} 1 : \frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)} > \tau \\ \quad \text{and } \rho(\sigma_t) > \text{max}(\Sigma_{\sigma_{t-1}}^{intra}) \\ 0 : \text{otherwise} \end{cases} \quad (4)$$

if $\delta_l(\sigma_{t-1}) = 1$,

$$\delta(t, \tau) = \begin{cases} 0 : \frac{\rho(\sigma_t)}{\text{mean}(\Sigma_{\sigma_{t-1}}^l)} < \frac{1}{\tau} \\ \quad \text{and } \rho(\sigma_t) < \text{min}(\Sigma_{\sigma_{t-1}}^{inter}) \\ 1 : \text{otherwise} \end{cases} \quad (5)$$

Indeed, if σ_{t-1} was labeled as intra-cluster, labeling σ_t as inter-cluster is consistent when its perimeter is greater than the maximum perimeter of $\Sigma_{\sigma_{t-1}}^{intra}$. Analogously, if σ_{t-1} was labeled as inter-cluster, labeling σ_t as intra-cluster would be consistent if its perimeter would be smaller than the minimum perimeter of $\Sigma_{\sigma_{t-1}}^{inter}$.

IV. EXPERIMENTS

The proposed approach is tested on two 2D shapes datasets¹, namely *Aggregation* (7 clusters) and *Compound* (6 clusters). The datasets are normalized to variance 1 and mean 0 and the implementation relies on our C++ implementation of d -dimensional algorithm in [24].

A. About determining the threshold value and the first simplex

As the technique relies on the choice of parameter τ , we provide first the grounds for its choice for the sake of the next section demonstration. Second (respectively third) column of Figure 3 plot, for all adjacent simplices σ and σ' in a given dataset, $\frac{\rho(\sigma)}{\rho(\sigma')}$ ratio distribution, where $\delta_l(\sigma) = 0$ and $\delta_l(\sigma') = 0$ (resp. $\delta_l(\sigma) = 0$ and $\delta_l(\sigma') = 1$). Observations on the two datasets show that τ should be chosen in $[1.65, 3.48]$. 1.65 is the average value of the third quartiles of the 13 intra-cluster boxplots in Figure 3. 3.48 is the average value of the first quartiles of the intra / inter pairs of simplices. Accordingly, we set $\tau = 3.48$ for the rest of the experiments.

For the construction of the first Delaunay simplex, which is crossed by the first hyperplane α_0 , we select α_0 orthogonal to the dimension which presents the highest frequency, as the first simplex is labeled as intra-cluster. For *Aggregation*, we set $\{(x, y), x = 1.5\}$ as the hyperplane α_0 and $\{(x, y), x = -1.0\}$ for *Compound*.

B. The evaluation of the technique: execution on synthetic datasets

The proposed technique yields the DT of the a point cloud for which a label is assigned to each simplex. The triangulation of *Aggregation* is composed of 1555 simplices. Our algorithm labels 1443 of them as intra-cluster and 111 as inter-cluster. Figure 3 illustrates the result of the proposed technique and

¹<http://cs.joensuu.fi/sipu/datasets/>

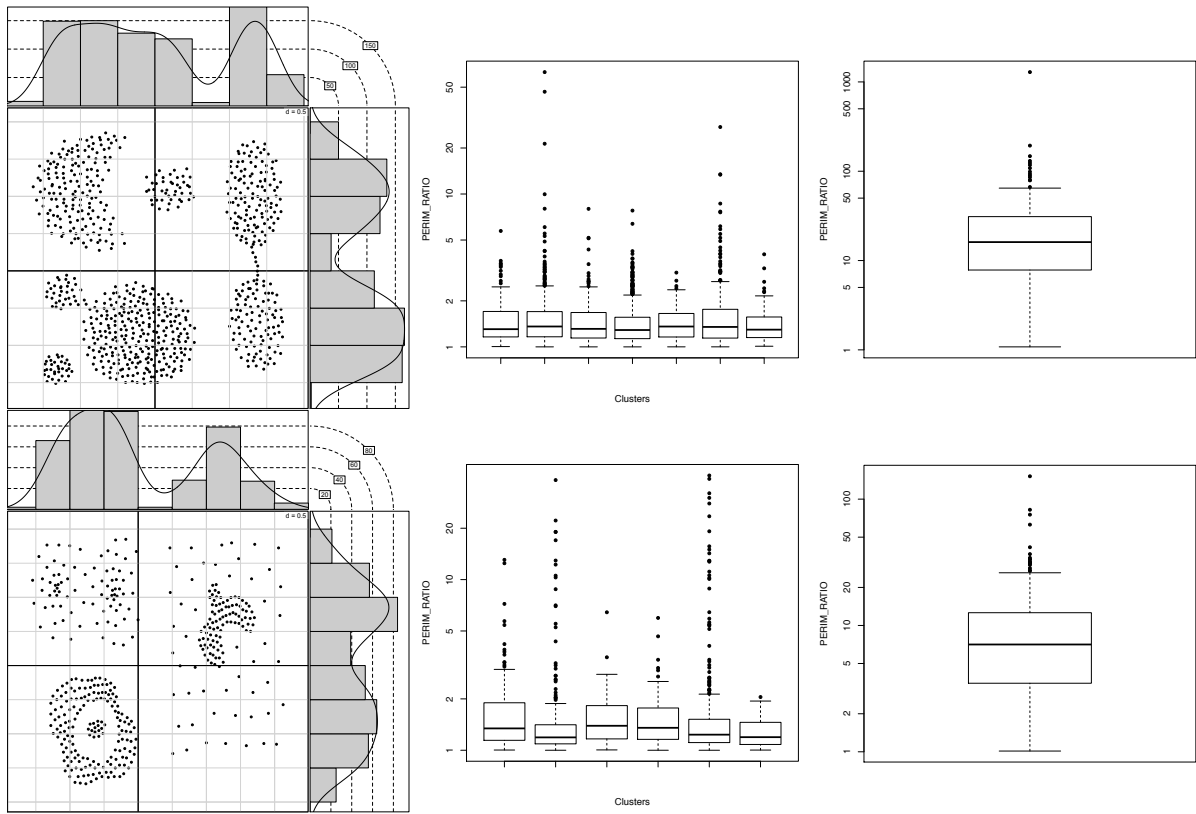


Fig. 2. **Datasets statistics.** **First column:** Scatterplots of the two tested datasets and analysis of their distribution along each dimension. **Second column:** Boxplots of the intra-cluster adjacent simplices perimeter ratio. **Third column:** Boxplots of the adjacent simplices that share a face that belongs to the boundary of the clusters

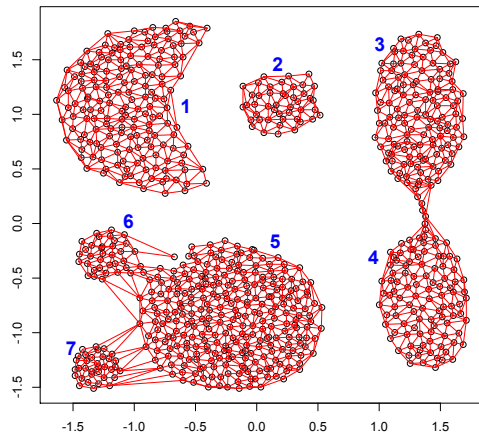


Fig. 3. **Aggregation dataset result.** The execution of the technique over Aggregation dataset revealed 4 connected components.

displays the intra-cluster simplices. The algorithm detects four connected components: $\{\{1\}, \{2\}, \{3,4\}, \{5,6,7\}\}$. Among those connected components, cluster 1 and 2 are correctly disconnected unlike cluster 3 and 4 which are connected. Clusters 5, 6 and 7 are merged in the same connected component. Following a density-based definition of clusters, connection of clusters 3 and 4 is topology preserving in the sense that there exist points that link these two clusters, thus no gap is

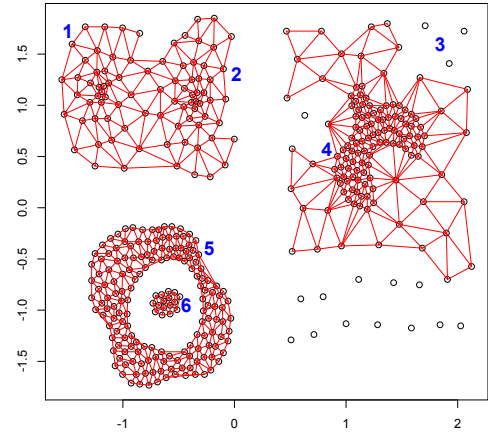


Fig. 4. **Zahn dataset statistics.** The execution of the technique over Compound dataset revealed 4 connected components disregarding the singletons.

detected in the perimeter distribution between them. The same argument applies to clusters 5 and 6. However, clusters 5 and 7 are wrongly connected by mislabeled simplices. The algorithm fails to disconnect these clusters, regarding the ground-truth labeling, as it detects a denser zone in this place of the point cloud.

The triangulation of *Compound* is composed of 777 sim-

plices. Our algorithm labels 632 of them as intra-cluster and 145 as inter-cluster. The algorithm detects four non-singleton connected components: $\{\{1,2\}, \{3,4\}, \{5\}, \{6\}\}$ (cf. Figure 4). Clusters 1 and 2 are connected as the simplices separating them is merely as large as the one within them. The algorithm fails at disconnected nested clusters 3 and 4 as there is no break of continuity when passing from cluster 4 to 3. However, it successfully disconnects nested clusters 5 and 6 as the inter-cluster simplices between them are larger than their intra-cluster ones.

V. DISCUSSION AND CONCLUSION

In this paper, we presented an original solution to the problem of cluster detection. The proposed technique adapts an algorithm from the computational geometry field to the problem of clustering. Precisely, this technique enriches an incremental *divide-and-conquer gift-wrapping* algorithm that constructs the DT of a point cloud by notably detecting the border of existing clusters. The technique proposes an original *one-pass* cluster detection strategy as, in opposition to the other techniques in the literature, it detects clusters as along with the *proximity graph* construction. This technique is based on the assumption that clusters are denser parts of the space which are separated by way sparser space. This hypothesis is reformulated in terms of perimeter of simplices and gaps in their distribution. The experiments notably showed that the proposed technique failed at detecting nested clusters that present a great difference in their perimeter value distribution. As well, the technique might perform poorly if the first Delaunay simplex is not actually within a cluster and if a proper τ is not found.

However, the experiments showed that the proposed technique manages well with arbitrary shaped clusters. Even if the number of ground-truth clusters is not found, it notably detected the hull of the clusters and the topological relationship among these. Moreover, the algorithm manages to deal with clusters presenting locally heterogeneous distribution of perimeters by maintaining the *short-term memory* of the clusters that gave to a given simplex its label and the *long-term* when this information is not available.

Forthcoming works include providing a thorough investigation in order to transpose what the ground-truth defines as intra-clusters in terms of simplices perimeter to find a *golden* τ and propose an heuristic for the first simplex selection. Other future works include testing the proposed solution on higher dimensional datasets and, as the technique allows to detect the border of clusters, adapt it to the problem of incremental clustering.

REFERENCES

- [1] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, pp. 255–308, 2009.
- [2] J. Jaromczyk, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, 1992.
- [3] B. Delaunay, "Sur La sphère vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, pp. 793–800, 1934.
- [4] G. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern recognition*, vol. 12, pp. 261–268, 1980.
- [5] R. K. Gabriel and R. R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, 1969.
- [6] O. Boruvka, "O Jistém Problému Minimálním (About a Certain Minimal Problem) (in Czech, German summary)," *Práce Mor. Přírodoved. Spol. v Brne III*, vol. 3, 1926.
- [7] D. G. Kirkpatrick and J. D. Radke, "Framework for computational morphology," *Computational Geometry*, pp. 217–248, 1985.
- [8] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [9] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 16, no. 3, pp. 645–78, May 2005.
- [10] G. Karypis, E. Han, and V. Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *Computer*, 1999.
- [11] C. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 68–86, 1971.
- [12] F. Muhlenbach and S. Lallich, "A new clustering algorithm based on regions of influence with self-detection of the best number of clusters," *Data Mining, 2009. ICDM'09. Ninth ...*, 2009.
- [13] R. Jiamthaphaksin, J. Choo, and C.-s. Chen, "MOSAIC: Agglomerative Clustering with Gabriel Graphs," 2008.
- [14] V. Estivill-Castro and I. Lee, "AUTOCLUST: Automatic clustering via boundary extraction for mining massive point-data sets," *In Proceedings of the 5th International Conference on Geocomputation*, 2000.
- [15] D. Liu, G. V. Nosovskiy, and O. Sourina, "Effective clustering and boundary detection algorithm based on Delaunay triangulation," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1261–1273, Jul. 2008.
- [16] V. Estivill-Castro and M. E. Houle, "Robust Clustering of Large Georeferenced Data Sets," pp. 327–338, 1999.
- [17] O. Razafindramanana and G. Venturini, "Delaunay simplices pruning based clustering," *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.
- [18] F. Aurenhammer, R. Klein, and L. Der-Tsai, *Voronoi Diagrams and Delaunay Triangulations*. World Scientific Publishing Company, 2013.
- [19] P. Cignoni and C. Montani, "A Merge-first divide & conquer algorithm for Ed Delaunay triangulations," *Intern Rep C*, 1992.
- [20] P. Su and R. S. Drysdale, "A comparison of sequential Delaunay triangulation algorithms," *Computational Geometry*, pp. 1–24, 1997.
- [21] A. Bowyer, "Computing Dirichlet tessellations," *The Computer Journal*, vol. 24, no. 2, pp. 162–166, 1981.
- [22] D. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The computer journal*, vol. 24, no. 2, pp. 167–172, 1981.
- [23] R. A. Dwyer, "Higher-dimensional voronoi diagrams in linear expected time," *Discrete & Computational Geometry*, vol. 6, no. 1, pp. 343–367, Dec. 1991.
- [24] P. Cignoni, "DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed," *Computer-Aided Design*, vol. 30, no. 5, pp. 333–341, Apr. 1998.
- [25] O. Razafindramanana, F. Rayar, and G. Venturini, "Alpha*-Approximated Delaunay Triangulation Based Descriptors for Handwritten Character Recognition," in *12th International Conference on Document Analysis and Recognition*, Washington D.C., USA, 2013.