ELSEVIER

Pattern Recognition Letters
journal homepage: www.elsevier.com

A fast persistence-based segmentation of noisy 2D clouds with provable guarantees

Vitaliy Kurlin[a,**]

[a]*Microsoft Research, 21 Station Road, Cambridge CB1 2FB and Mathematical Sciences, Durham University, Durham DH1 3LE, UK.*

ABSTRACT

We design a new fast algorithm to automatically segment a 2D cloud of points into regions. The only input is a dotted image without any extra parameters, say a scanned black-and-white map with almost closed curves or any image with detected edge points. The output is a hierarchy of segmentations into regions whose boundary contours have a long enough life span (persistence) in a sequence of nested neighborhoods of the input points. We give conditions on a noisy sample of a graph, when the boundaries of resulting regions are geometrically close to all original cycles in the unknown graph.

1. Introduction: the problem, summary and applications

1.1. Problem: automatic segmentation of a noisy 2D cloud

Recognising closed contours is a primary phenomenon according to Gestalt laws of perception. Humans can easily form closed loops from incomplete contours. Closed contours bound holes or semantic regions whose extraction is needed for higher level image understanding. So a robust location of holes and their boundary contours is a key part of the low level vision.

The problem is most similar to Saund (2003), who found perceptually closed contours using weights of various quality criteria. Completing closed contours is equivalent to finding regions bounded by these contours. The segmentation problem will be solved in the hardest form without using extra input parameters: *given only a noisy sample C of a graph $G \subset \mathbb{R}^2$, reconstruct closed contours approximating all cycles of G.*

The input is a *cloud* C, which is a finite set of points with any real coordinates in the plane $\mathbb{R}^2$, see Fig. 1. Such a cloud can be a noisy sample or a scan of a hand-drawn sketch or an artistic drawing. The input can also be a binary black-and-white image or a sparse cloud of points without any pixel connectivity.

The output is a hierarchy of segmentations into regions whose boundaries are non-self-intersecting cycles of straight edges connecting points in a given cloud. Fig. 1 shows a cloud C and the most persistent segmentation, which was found by using only C without parameters. All regions bounded by resulting contours are painted with random colors for simplicity.

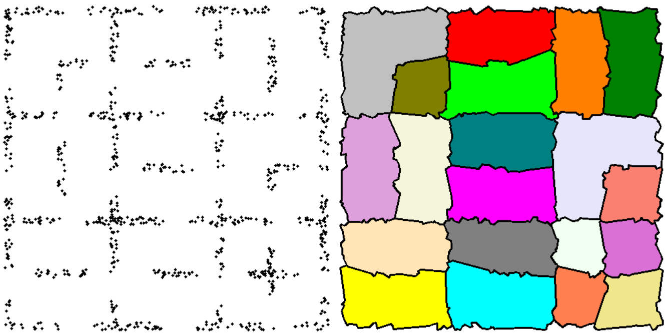

Fig. 1. Input: a noisy cloud C randomly sampled near a graph $G \subset \mathbb{R}^2$. Output: randomly painted most persistent regions (best viewed in colour).

The algorithm finds the full hierarchy of segmentations in time $O(n \log n)$ for a cloud C of n points. The boundary contours in a 1st segmentation of the hierarchy have approximation guarantees for noisy samples of a graph $G \subset \mathbb{R}^2$ in Theorem 9.

In addition to the motivations of Saund (2003) we give four more applications for locating holes and their boundaries.

Auto-closure of polygons. A typical difficulty for users of graphics software is to accurately match endpoints of a polygonal line for painting a resulting region. The suggested approach resolves this difficulty without asking users for extra input.

Hierarchical segmentation. When there is no single ideal segmentation as in most natural images, the algorithm can propose several most likely segmentations, which are rigorously quantified by their stable-under-noise topological persistence.

[**]Corresponding author: Tel.: +44-1913343081; fax: +44-1913343051; *e-mail:* `vitaliy.kurlin@gmail.com` (Vitaliy Kurlin)

Automatic colorization. Artistic drawings are line sketches that often contain gaps, but may give an impression of closed contours. We may enhance these black-and-white drawings by automatically painting regions that appear visually closed.

Map reading. Conventional paper maps contain many level set curves. These curves are often split into disjoint arcs by labels showing actual heights above the sea level. Hence auto-completion of contours can help faster digitize paper maps.

1.2. Related Work on Topology-based Image Segmentation

We review only methods that are close to the proposed one.

Pixel-based segmentation is a traditional approach to find boundary contours in a pixel-based image. Such a segmentation usually minimizes an energy function that contains a cost of assigning a single label and a cost of assigning different labels to neighboring pixels. The resulting minimization problem is often NP-hard. Including higher-order potential between more than two pixels has even larger computational costs and still encodes only local properties. Chen et al. (2011) suggested the first binary segmentation with global topological constraints, e.g. when a foreground object is connected and has no holes.

Extra parameters are essentially needed for many algorithms including Chernov and Kurlin (2013) and the image segmentation by Letscher and Fritts (2007). They start with a Delaunay triangulation of a point cloud C as in our first step. Then small triangles merge to form persistent regions using two threshold parameters: α for the radius of disks centered at the points of C, and p for the desired level of persistence. We substantially improve this method to avoid any input parameters and prove guarantees by using the stability of 1D persistence.

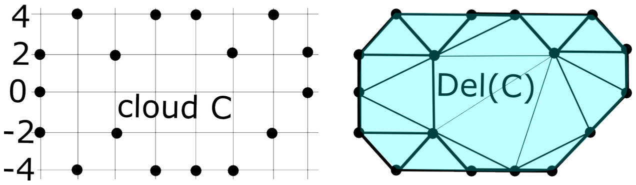

Fig. 2. A cloud C and its (non-unique) Delaunay triangulation Del(C)

Fast 1D persistence for offsets of 2D point clouds can be computed in time $O(n \log n)$ by the standard algorithm of Attali et al. (2009). This approach was applied by Kurlin (2014b) for counting topologically persistent holes in noisy clouds. However, the 1D persistence diagram contains only unstructured data. A segmentation of a cloud requires more information about adjacency relations of persistent regions. We substantially extend the algorithm from Attali et al. (2009) by adding a new sophisticated data structure Map(α), which allows one to merge all regions into most persistent ones, see section 4.

Homology inference conditions were obtained in many cases to guarantee a correct topological reconstruction from a sample C. In addition to topological guarantees, for the first time Theorem 9 provides a lower bound for the unknown noise level ε and gives conditions on a graph G when reconstructed contours are geometrically 2ε-close to true cycles of G.

1.3. Contributions: Parameterless Algorithm and Guarantees

The key differences between the new automatic solution and all the past segmentation methods above are the following.

- We solve the harder problem of completing closed contours or segmenting a cloud of points *without any extra parameters*.
- Input points can have any real coordinates in $\mathbb{R}^2$, the algorithm works for feature points in images at any *subpixel resolution*.
- The algorithm is unsupervised and outputs a hierarchy, where one can get a segmentation with a *required number of regions*.
- The quality of segmentations is measured by the *stable-under-noise persistence*, which leads to guarantees in Theorem 9.

No scale parameters are needed, because a given cloud is analyzed for all values of a radius α. The 1st output consists of those contours whose persistence is above a 1st widest gap in a persistence diagram, see Definition 2 and 3. The 2nd output has contours with a persistence above the lowest of the first 2 widest gaps etc. If exactly k regions are needed, one can select contours corresponding to k cycles with highest persistence.

Data-driven measurements are used for quantifying persistence of contours. If a point cloud C lives in a metric space, we have only a distance function for studying a shape of C. So a natural representation of such a shape is the *α-offset* C^α, which is the union of disks with a radius $\alpha > 0$ and centers at all points of C. This complicated α-offset C^α continuously deforms to the simpler α-complex $C(\alpha)$ for any α, see Fig. 3 and Definition 1.

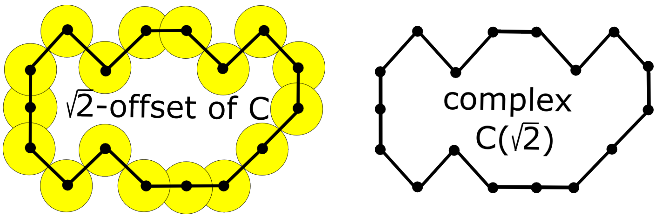

Fig. 3. The α-offset C^α deforms to the α-complex $C(\alpha)$ for $\alpha = \sqrt{2}$.

The persistence of a contour is its life span in the sequence of nested offsets C^α. When α is increasing, a contour is *born* at some α = birth and dies in a larger offset at α = death. The *persistence* is death − birth, see Definition 2. For C in Fig. 2, the 1st contour is born at $\alpha = \sqrt{2}$ and persists until all triangles of Del(C) are covered by disks of a larger radius α, see Fig. 3.

Here is a high-level description of the key contributions.

- **Near linear time**: for any n points in $\mathbb{R}^2$, a hierarchy of segmentations is computed in time $O(n \log n)$, see Theorem 7.
- **Guarantees**: if a cloud C is densely sampled from a good enough graph $G \subset \mathbb{R}^2$, all contours of G can be geometrically approximated by using only the cloud C, see Theorem 9.
- **Stability**: the output is globally stable under noise in a cloud C, namely remains in a small offset of C, see Corollary 10.
- **Experiments** on synthetic and natural images in section 6 confirm that the results are robust even for really large noise.

2. Persistent homology of α-complexes $C(\alpha)$ in the plane

We briefly introduce key concepts, see details in (Kurlin, 2015, Appendix A). The α-offsets C^α have complicated shapes, but continuously deforms to α-complexes, which are substructures of a Delaunay triangulation, see Edelsbrunner (1995).

Definition 1 (α-complexes $C(\alpha)$ and holes). *A* Delaunay triangulation $\mathrm{Del}(C)$ *on a cloud $C \subset \mathbb{R}^2$ consists of all triangles whose vertices are from C and whose circumcircles enclose no points of C. For any scale $\alpha > 0$, the* α-complex $C(\alpha)$ *is obtained from* $\mathrm{Del}(C)$ *by removing all open edges longer than 2α and all open triangles with circumradii more than α. A* hole *of $C(\alpha)$ is a bounded connected component of $\mathbb{R}^2 - C(\alpha)$. The boundaries of all holes are* boundary contours *of $C(\alpha)$.*

If $\alpha > 0$ is small, the α-complex $C(\alpha)$ consists of all isolated points of C. For any large enough α, the complex $C(\alpha)$ is $\mathrm{Del}(C)$. So the α-complex $C(\alpha)$ is built on isolated points of C by adding edges and triangles at the following critical values:

- an edge between points $p_i, p_j \in C$ is added at $\alpha = \frac{1}{2}d(p_i, p_j)$;
- an *acute* triangle (that has all angles less than $\frac{\pi}{2}$) is added at the critical value α equal to the circumradius of the triangle;
- a non-acute triangle T is added to $C(\alpha)$ at the scale α that is equal to the half-length of the largest side in the triangle T.

All α-complexes form the *filtration* $\{C(\alpha)\}$ that is a nested sequence $C = C(0) \subset \ldots \subset C(\alpha) \subset \ldots \subset C(+\infty) = \mathrm{Del}(C)$.

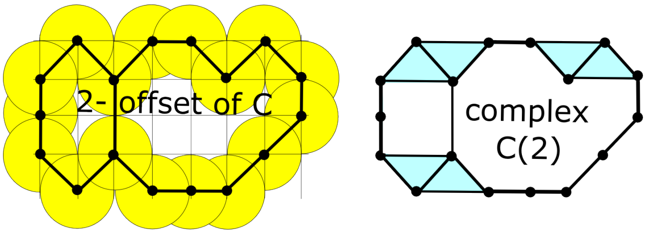

Fig. 4. The α-offset C^α deforms to the α-complex $C(\alpha)$ for $\alpha = 2$.

The summary of topological changes in the filtration $\{C(\alpha)\}$ is quantified by life spans (from birth to death) of holes. For the cloud C in Fig. 2, the first hole is born at the scale $\alpha = \sqrt{2}$ in Fig. 3 and splits into 2 smaller holes at $\alpha = 2$, see Fig. 4.

The left-hand-side hole in Fig. 4 is a square, which shrinks to the triangle with sides $4, 2\sqrt{5}, 2\sqrt{5}$ in $C(\sqrt{5})$, see Fig. 5. The triangle has the circumradius $R_2 = 2.5$ and enters $C(\alpha)$ at the scale $\alpha = 2.5$, hence the hole persists over $2 \le \alpha < 2.5$.

The right-hand-side hole in Fig. 4 splits again at $\alpha = \sqrt{5}$ giving birth to the 3rd triangular hole with sides $2\sqrt{2}, 2\sqrt{5}, 2\sqrt{5}$. This acute triangle has the circumradius $R_3 = \frac{5}{3}\sqrt{2} \approx 2.357$ and enters $C(\alpha)$ at $\alpha = R_3$, so the 3rd hole persists over $\sqrt{5} \le \alpha < R_3$. The remaining hole in $C(2.5)$ contains the Delaunay triangle with the largest circumradius $R_1 = \frac{5}{7}\sqrt{26} \approx 3.642$, see 2. So the most persistent hole lives over $\sqrt{2} \le \alpha < R_1$.

The 1-dimensional homology $H_1(S)$ counts holes of a complex $S \subset \mathbb{R}^2$. Formally, $H_1(S)$ is a vector space generated by 1-dimensional cycles that enclose holes of S. If a connected complex $S \subset \mathbb{R}^2$ consists of V vertices, E edges, F triangles, then $\chi(S) = V - E + F$ is the *Euler characteristic* and $\beta_1(S) = 1 - \chi(S)$ is the first *Betti* number equal to the number of holes of S.

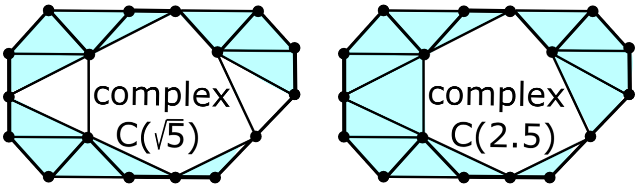

Fig. 5. The α-complexes $C(\alpha)$ of the cloud C in Fig. 2 for $\alpha = \sqrt{5}, 2.5$.

The complex $C(\sqrt{2})$ in Fig. 3 has $V = E = 17$, $F = 0$, so $\beta_1 = 1$. The vector space $H_1(S)$ of linear combinations with coefficients in the field $\mathbb{Z}_2 = \{0, 1\}$ is determined by $\dim H_1(S) = \beta_1(S)$, so $H_1(S) = \mathbb{Z}_2^{\beta_1(S)}$. The α-complex $C(\sqrt{5}) \subset \mathbb{R}^2$ in Fig. 5 has 3 holes and $H_1(C(\sqrt{5})) = \mathbb{Z}_2^3$.

Since any α-offset C^α continuously deforms to the α-complex $C(\alpha)$, they have and have the same homology H_1, see (Kurlin, 2015, Appendix A). In practice, $H_1(C^\alpha)$ is unstable under perturbations of a cloud C. This instability is resolved by the more advanced concept of persistent homology that tracks changes in the homology $H_1(C^\alpha)$ across all scales $\alpha > 0$.

Inclusions $C(\alpha_i) \subset C(\alpha_j)$ for any $\alpha_i < \alpha_j$ induce a sequence of linear maps $H_1(C(\alpha_i)) \to H_1(C(\alpha_j))$ decomposable into a sum of elementary sequences over life spans of each hole. The α-complexes for the cloud C in Fig. 2 induce $H_1(C(0)) = 0 \to$

$$\to H_1(C(\sqrt{2})) = \mathbb{Z}_2 \to H_1(C(2)) = \mathbb{Z}_2^2 \to H_1(C(\sqrt{5})) = \mathbb{Z}_2^3 \to$$

$$\to H_1(C(R_3)) = \mathbb{Z}_2^2 \to H_1(C(R_2)) = \mathbb{Z}_2 \to H_1(C(R_1)) = 0,$$

which decomposes as a sum of 3 sequences $0 \to \mathbb{Z}_2 \xrightarrow{\mathrm{id}} \mathbb{Z}_2 \to 0$ over 3 life spans $\sqrt{2} \le \alpha < R_1$, $2 \le \alpha < R_2$ and $\sqrt{5} \le \alpha < R_3$.

Definition 2 (persistence diagram $\mathrm{PD}\{C(\alpha)\}$**).** *A homology class $\gamma \in H_1(C(\alpha_i))$ is* born *at a scale α_i =* birth(γ) *if γ is not in the image of the map $H_1(C(\alpha)) \to H_1(C(\alpha_i))$ for any $\alpha < \alpha_i$. The class γ* dies *at α_j =* death(γ) *$\ge \alpha_i$ when the image of γ under $H_1(C(\alpha_i)) \to H_1(C(\alpha_j))$ merges into the image of $H_1(C(\alpha)) \to H_1(C(\alpha_j))$ for some smaller scale $\alpha < \alpha_i$. So by the* elder rule *a younger hole born at α_i dies by merging another older hole, which was born earlier at $\alpha < \alpha_i$. All pairs* (birth, death) *form the* persistence diagram $\mathrm{PD}\{C(\alpha)\} \subset \mathbb{R}^2$.

The filtration $\{C(\alpha)\}$ of α-complexes for a cloud C in Fig. 2 has the persistence diagram in Fig. 6 containing 3 dots (birth, death) for the 3 life spans of holes or their boundaries.

Points near the diagonal have a low persistence death − birth and are considered as noise. If $\mu > 1$ different holes are born and die at the same scales, the corresponding dot (birth, *death*) has the *multiplicity* μ. It is safe to add to a persistence diagram all diagonal points (α, α) with infinite multiplicity, because they have persistence 0, so $\mathrm{PD}\{C(\alpha)\}$ is a multi-set of dots.

Stability of persistence from Cohen-Steiner et al. (2007) says that perturbing a cloud C within its ε-offset similarly perturbs $\mathrm{PD}\{C(\alpha)\}$ up to ε, see (Kurlin, 2015, Appendix A). After the introduction above, we define the new concept of diagonal gaps in persistence leading to a hierarchy of segmentations.

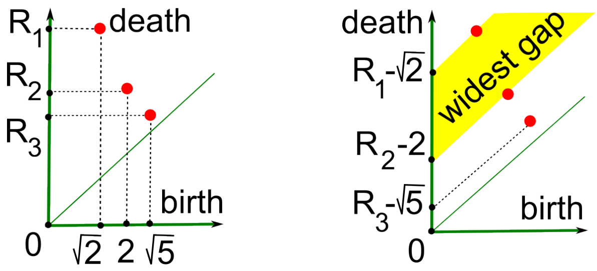

Fig. 6. The persistence diagram of $\{C(\alpha)\}$ for the cloud C in Fig. 2.

Definition 3 (diagonal gaps and subdiagram PD_k). *For the filtration $\{C(\alpha)\}$ of a cloud $C \subset \mathbb{R}^2$, a diagonal gap is a largest strip $\{a < y - x < b\}$ that has pairs $\mathrm{PD}\{C(\alpha)\}$ on the boundary lines, but not inside the strip, see Fig. 6. For any integer $k \geq 1$, the k-th* widest diagonal gap *has the k-th largest vertical width $b - a$. If there are several widest gaps with the same width, we choose the lowest gap. The* k-th subdiagram $\mathrm{PD}_k \subset \mathrm{PD}\{C(\alpha)\}$ *consists of all dots above the lowest of the first k widest gaps.*

Each k-th widest gap separates dots $(\mathrm{birth}, \mathrm{death})$ with a higher persistence death − birth from dots with a lower persistence. Definition 4 introduces a new structure of persistent regions and their boundary contours directly on a cloud C.

Definition 4 (k-th segmentation of C into persistent regions). *For a finite cloud $C \subset \mathbb{R}^2$, consider all m_k dots* $(\mathrm{birth}, \mathrm{death})$ *in the k-th subdiagram* $\mathrm{PD}_k\{C(\alpha)\}$. *Each dot* $(\mathrm{birth}, \mathrm{death})$ *corresponds to a region bounded by a closed cycle whose homology class was born at α = birth. For each triangle T outside these m regions in* $\mathrm{Del}(C)$*, we merge T with its adjacent region along a longest edge of T. After all mergers, the final m_k regions form the* k-th segmentation *of C into* persistent regions.

The persistence diagram in Fig. 6 has only one dot $(\sqrt{2}, R_1)$ above the 1st widest gap. The boundary of the corresponding region is the closed contour $C(\sqrt{2})$ in Fig. 3. The remaining 3 right-angled triangles outside the complex $C(\sqrt{2})$ merge with the external region. By Definition 4 the 1st segmentation of C consists of the single region enclosed by $C(\sqrt{2})$, see Fig. 9.

The information about merged regions is not contained in the classical 1D persistence diagram consisting of only dots $(\mathrm{birth}, \mathrm{death})$. Hence the new algorithm in section 4 substantially extends the standard persistence computation to get a hierarchy of persistent structures directly on a cloud C.

3. Duality between α-complexes and α-graphs in the plane

We analyze the evolution of contours in the filtration of 2D α-complexes $C(\alpha)$ using the simpler filtration of 1D α-graphs $C^*(\alpha)$ that are dual to $C(\alpha)$. Let us associate a node v_i to every triangle in $\mathrm{Del}(C)$, call the external region of the triangulation $\mathrm{Del}(C)$ also a 'triangle' and represent it by an extra node v_0. So v_i are abstract nodes shown as small red circles in Fig. 7.

Definition 5 (α-graphs $C^*(\alpha)$ of a cloud C). *Let the metric graph C^* dual to* $\mathrm{Del}(C)$ *have the nodes $v_0, v_1, \ldots, v_k$ and edges of a length $\frac{1}{2}d_{ij}$ connecting nodes v_i, v_j such that the corresponding triangles in* $\mathrm{Del}(C)$ *share a longest side of the length d_{ij}. The α-graph $C^*(\alpha)$ is obtained from C^* by removing all edges not longer than α. Any isolated node v (except v_0) is removed from the graph $C^*(\alpha)$ if the corresponding triangle T_v in* $\mathrm{Del}(C)$ *is not acute or has a small* circumradius $\mathrm{rad}(v) \leq \alpha$.

The smallest graph $C^*(+\infty)$ is the isolated vertex v_0 corresponding to the external region of $\mathrm{Del}(C)$. When α drops from 2.5 to $\sqrt{5}$ in Fig. 7, two isolated nodes v_2 and v_3 enter $C^*(\alpha)$, because $\mathrm{rad}(v_2) = R_2 = 2.5$, $\mathrm{rad}(v_3) = R_3 = \frac{5}{3}\sqrt{2} > \sqrt{5}$. However, these nodes remain isolated in $C^*(\sqrt{5})$ because all sides of their triangles have half-lengths not longer than $\sqrt{5}$.

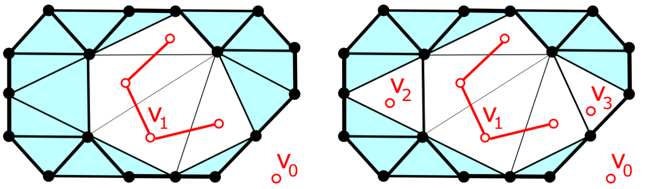

Fig. 7. The complex $C(\alpha)$ and graph $C^*(\alpha)$. Left: $\alpha = 2.5$. Right: $\alpha = \sqrt{5}$.

The ascending filtration of a Delaunay triangulation $\mathrm{Del}(C)$ by α-complexes gives rise to the descending filtration of α-graphs $C^* = C^*(0) \supset \ldots \supset C^*(\alpha) \supset \ldots \supset C^*(+\infty) = \{v_0\}$. Each connected component of $C^*(\alpha)$ has the corresponding region enclosed by a boundary contour of the complex $C(\alpha)$.

For instance, if $C(\alpha)$ contains a triangular cycle, but not the enclosed triangle T_v, then the circumradius $\mathrm{rad}(v) > \alpha$ and the corresponding node v belongs to $C^*(\alpha)$. So there is a 1-1 correspondence between all isolated nodes (except v_0) of the graph $C^*(\alpha)$ and all triangular boundaries of $C(\alpha)$. We extend this duality to all components of $C^*(\alpha)$ and all boundaries of $C(\alpha)$.

Lemma 6 (duality between C and C^*). *For any scale $\alpha > 0$, all components of the α-graph $C^*(\alpha)$ are in a 1-1 correspondence with all boundary contours of the α-complex $C(\alpha)$.*

Proof. When α is decreasing, the birth of a boundary contour in $C(\alpha)$ means that a small hole appears around the center (of the circumcircle) of an acute triangle $T_v \subset \mathrm{Del}(C)$. By Definition 5 at the same time the isolated node v corresponding to the triangle T_v enters the graph $C^*(\alpha)$ as a new component.

The death of a boundary contour in $C(\alpha)$ means that the contour is torn at its longest edge e for α = half-length of e. If the edge e is shared by triangles T_u, T_v, the corresponding nodes u, v become linked, their components merge in $C^*(\alpha)$.

So there is a 1-1 correspondence between births of contours in $C(\alpha)$ and components in $C^*(\alpha)$, and similarly between their deaths. In general, for any fixed value of α, each component of $C^*(\alpha)$ containing nodes $v_1, \ldots, v_k$ is dual to the contour going along the boundary of the union $T_1 \cup \ldots \cup T_k \subset \mathrm{Del}(C)$ of the triangles represented by the nodes $v_1, \ldots, v_k \in C^*(\alpha)$. □

When α is decreasing from $+\infty$ to 0, the α-complex $C(\alpha) \subset \mathbb{R}^2$ is shrinking, while $C^*(\alpha)$ is growing by Duality Lemma 6. Initially, $C(+\infty) = \mathrm{Del}(C)$ and we show all triangles of $\mathrm{Del}(C)$ in blue. If a triangle disappears from $C(\alpha)$ at a critical value α, for all smaller α we put a small red circle in this triangle.

4. New data structure Map(α) for segmenting 2D clouds

The nice algorithm of Attali et al. (2009) for a fast 1D persistence requires an essential extension for segmenting 2D clouds, because the 1D persistence diagram PD$\{C^\alpha\}$ has no information about adjacency of required persistent regions. Informally, we will merge all noisy dots in PD$\{C^\alpha\}$ with a small number of dots above the widest gap, which leads to a final segmentation.

Nodes of the α-graph $C^*(\alpha)$ with attributes are stored in the array Forest(α). Components of $C^*(\alpha)$ describing persistent regions are in the array Map(α). Briefly, the algorithm maintains a union-find structure on Forest(α) and updates adjacency relations in Map(α) when one region mergers another one.

By Definition 5 all nodes of $C^*(\alpha)$ are in a 1-1 correspondence $v \leftrightarrow T_v$ with all triangles of a Delaunay triangulation Del(C), where the external region of Del(C) is also called a 'triangle'. We call a node $u \in$ Forest(α) blue if $T_u \in C(\alpha)$, otherwise u is red. Initially for a large α, all nodes are isolated and blue. When α is decreasing, the nodes start turning *red* and join each other to form *red* components of the α-graph $C^*(\alpha)$.

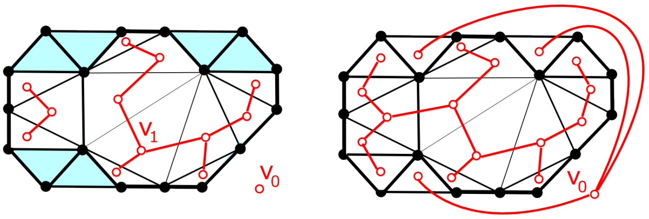

Fig. 8. The complex $C(\alpha)$ and graph $C^*(\alpha)$. Left: $\alpha = 2$. Right: $\alpha = \sqrt{2}$.

Merging two components. When two nodes become linked and their red components merge, a younger component dies. Since α is decreasing, a component is younger if its first node was born at a smaller scale α than the older component. The older component survives by the *elder rule* (Edelsbrunner and Harer, 2010, p. 150), which aims to maximize persistence.

The new larger red component contains the dead nodes from the younger component and the *live* nodes (or corresponding *live* triangles) from the older component. Since α is decreasing, any younger component dies at a smaller value of α than its birth. So the persistence of a red component in $C^*(\alpha)$ and the corresponding boundary contour in $C(\alpha)$ is birth − death > 0.

In the structure Forest(α) any node v has these attributes:

- birth(v) = $\sup\{\alpha \mid v \in C^*(\alpha)\} = \sup\{\alpha \mid T_v \subset C(\alpha)\}$;
- uplink(v) is a unique parent of the node v in Forest(α);
- height(v) is the height of Tree(v) going down from v;
- Live(v) is the list of all triangles that are alive in Tree(v);
- bar(v) is the index of the region in Map(α) containing T_v.

For each acute triangle T_v in Del(C), the corresponding node v has birth(v) = circumradius rad(v) of T_v. For any non-acute triangle T_v, the node v is linked to an existing component when α is the half-length of the longest edge of T_v, so T_v merges with its neighbor. Starting from any node v, we come to root(v) by going up along uplinks until the node root(v) is a self-parent.

Then u, v belong to the same component of Forest(α) if and only if root(u) = root(v) as in a union-find structure. When we need to join nodes u, v from different components, we actually join their roots by adding a shorter tree to a taller tree.

Any root keeps the important information about its tree. For instance, the birth time for any node v is extracted as birth(root(v)). To justify the notation *bar*, any dot (birth, death) $\in \mathbb{R}^2$ can be considered as the interval [birth, death) $\subset \mathbb{R}$. These intervals or bars form a *bar code* equivalent to PD. A value bar(v) = k means that the triangle T_v is in the region whose boundary contour was the k-th to die.

Structure Map(α) **of persistent regions.** Any region R consists of triangles whose corresponding nodes $v_1, \dots, v_k$ form a connected component of $C^*(\alpha)$, and R has these attributes:

- ind(R) is the index of the region R in the array Map(α);
- birth(R) = scale α when a 1st triangle enters the region R;
- death(R) = α when R mergers with an older superior region;
- Core(R) is a list of nodes whose triangles form the region R;
- sup(R) is the index of the superior region merged with R.
- heir(R) is the node $v \notin$ Core(R) adjacent to a node $u \in$ Core(R) such that linking u, v mergers R and its superior region (of v);

The input is a set of n points given by real coordinates $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$. We start by finding the Delaunay triangulation Del(C) in time $O(n \log n)$ with $O(n)$ space.

Initialization. We set birth(v_0) = $+\infty$ for the external node $v_0 \in$ Forest(α). After finding Del(C), we go through each triangle T_v of Del(C) and set the birth of the corresponding node $v \in$ Forest(α) as the circumradius rad(v) for acute T_v and birth(T_v) = 0 for non-acute T_v. All bar(v) have the initial value 0 meaning that the bar indices are undefined. All arrays Live(v) and Map(α) are empty. We sort all edges of the triangles of Del(C) in the decreasing order starting from the longest edge of a length d. We start from the initial largest value $\alpha = \frac{1}{2}d$.

The 'while' loop goes through each edge e of Del(C) in the decreasing order of length until Forest(α) becomes connected. Let the edge e be shared by adjacent triangles T_u, T_v. If α is going down through the critical value equal to the half-length of e, we link the corresponding nodes u, v in the α-graph $C^*(\alpha)$. This addition doesn't affect Forest(α) if u, v were already connected, namely they have a common root as in Case 1 below.

It is possible that one of the nodes, say u, was not included in Forest(α) at the initialization stage, because the corresponding triangle T_u is not acute, so u was blue. In this Case 2 discussed below we link the single node u to the red component of v.

If the nodes u, v are in different trees of Forest(α), we merge these trees by linking their roots in Case 3. Final Case 4 below studies the exception when both triangles T_u, T_v are not acute. Then T_u, T_v are right-angled and have the common hypotenuse e, otherwise T_v is enclosed by the circumcircle of T_u, which is forbidden in a Delaunay triangulation Del(C) by Definition 1.

At the end Map(α) will contain adjacency relations of all regions in addition to the 1D persistence diagram PD$\{C(\alpha)\}$ as pairs (birth, death). Each region R will have the index sup(R) of its adjacent more persistent region that merged with R.

Case 1: the edge e has the same region on both sides, namely the neighboring triangles T_u, T_v sharing the edge v have the same root root(u) = root(v). This value of α is not critical, because both nodes u, v were already connected in Forest(α).

Case 2: e is the longest edge of a non-acute triangle T_u and another triangle T_v with $v \in$ Forest(α). We find root(v) and add u to Forest(α) setting uplink(u) = root(v), birth(u) = birth(root(v)). We increase height(root(v)) by 1 only if it was 1. If bar(v) is defined, then the node v belongs to the already dead region $R \in$ Map(α) with ind(R) = bar(v). Hence u joins this region R and we set bar(u) = bar(v). If bar(v) is undefined, then T_v is a live triangle and we add u to Live(root(v)).

Case 3: both birth(root(u)), birth(root(v)) > 0. Then the two components of Forest(α) containing the nodes u, v merge. Assume that birth(root(u)) $\leq$ birth(root(v)), so the component of u is younger, hence dies. We create a new region R in Map(α), say with an index i, by setting birth(R) = birth(root(u)), death(R) = α (the current value equals the half-length of e). All nodes $w \in$ Live(root(u)) die and we set bar$(w) = i$ for them.

If bar(v) is already defined, the component of v has died and we can set sup(R) = bar(v). Otherwise the component of v will die later and we set heir$(R) = v$ remembering to update sup(R) using heir(R) after the 'while' loop is finished. Then we copy the list Live(root(u)) to Core(R) so that the region R knows all its nodes (triangles) that were alive just before R died.

Subcase 3a: height(root(u)) $\leq$ height(root(v)). Then we link root(u) of the shorter tree to root(v) of the taller tree to keep to maximum height of all trees in Forest(α) minimal, so root(v) becomes uplink(root(u)). If the heights were equal, then height(root(v)) jumps up by 1. All live triangles from Live(root(v)) are kept at the root of the new larger tree.

Subcase 3b: height(root(u)) $>$ height(root(v)). Then we link root(v) of the shorter tree to root(u) of the taller tree. Hence root(u) becomes uplink(root(v)), but height(root(u)) remains the same. We should keep all triangles from Live(root(v)) at the root of the new tree replacing Live(root(u)) by Live(root(v)). So it is important to save Live(root(u)) in Core(R) as we did.

Case 4: birth(root(u)) = 0 = birth(root(v)) means that both triangles T_u, T_v are not acute and share their longest edge. This is possible only if T_u, T_v are right-angled with a common hypotenuse. Hence u, v form a new red component of $C^*(\alpha)$.

The external region in Map(α)**.** When Forest(α) becomes connected, it remains to add to Map(α) the last entry R corresponding to the external region of Del(C). The last root v has the list Live(v) containing the node v_0. Similarly to Case 3, Live(v) is copied to Core(R) and we set bar(w) equal to the index ind(R) for any node $w \in$ Live(v), but heir(R) is not needed.

Initial segmentation. Each triangle from Del(C) contributes to a single component of Map(α), namely all lists Core(R) are disjoint. Hence Map(α) contains $m = O(n)$ entries and can be sorted in time $O(n \log n)$ in the decreasing order of pers = birth − death. We output the initial segmentation where all triangles from Core(R) have a color associated with R. The 1st picture in Fig. 9 shows 3 initial regions corresponding to all 3 dots in the 1D persistence diagram PD$\{C^\alpha\}$ from Fig. 6.

Algorithm 1 Build Map(α) of persistent regions in a cloud C

```
Input: a cloud $C$ of $n$ points $(x_1, y_1), \dots (x_n, y_n)$
Compute Del$(C)$ with $k$ triangles on the $n$ points of $C$
Sort edges of Del$(C)$ in the decreasing order of length
Forest ← isolated nodes $v_0, \dots, v_k$ with all birth times 0
  except birth$(v_0)$ ← $+\infty$ and for each acute triangle $T_v \subset$
  Del$(C)$ we update birth$(v)$ ← circumradius of $T_v$
Set the number of links in Forest$(\alpha)$: $l \leftarrow 0$
while $l < k$ (stop when Forest$(\alpha)$ becomes a tree) do
    Take the next longest edge $e$, set $\alpha \leftarrow \frac{1}{2}$length$(e)$
    Find $u, v$ dual to the triangles $T_u, T_v$ that share $e$
    Find root$(u)$, root$(v)$ going along uplinks from $u, v$
    if root$(u)$ = root$(v)$ then Case 1: no changes, hence
        continue the loop without increasing the number $l$.
    end if
    if birth(root$(u)$) = 0 and birth(root$(v)$) > 0 then
        (there is also a symmetric case with $u, v$ swapped)
        Case 2: ($u$ is blue, $v$ is red) run Algorithm 2 below.
        The number of links $l \leftarrow l + 1$, continue the loop.
    end if
    if 0 < birth(root$(u)$) ≤ birth(root$(v)$) then
        ($u$ younger than $v$, another case with $u, v$ swapped)
        Case 3: components of $u, v$ merge, run Algorithm 3
        The number of links $l \leftarrow l + 1$, continue the loop.
    end if
    if birth(root$(u)$) = 0 = birth(root$(v)$) then
        Case 4: (the triangles $T_u, T_v$ are right-angled)
        Set birth$(u)$ = birth$(v)$ ← $\alpha$, uplink$(v)$ ← $u$,
        height$(u)$ ← height$(u)$+1, add the node $v$ to Live$(u)$
        The number of links $l \leftarrow l + 1$, continue the loop.
    end if
end while
Return array Map$(\alpha)$ of regions generated by Case 3
```

Algorithm 2 : link the node u to the component of the node v

```
Set uplink$(u)$ ← root$(v)$, birth$(u)$ ← birth(root$(v)$)
if height(root$(v)$) = 1 then height(root$(v)$) ← 2
end if
if bar$(v)$ is already defined then set bar$(u)$ ← bar$(v)$
else  Add $u$ to Live(root$(v)$) in the subtree at root$(v)$
end if
```

Algorithm 3 : the younger component becomes a region R

```
Create a new region $R \in$ Map$(\alpha)$ of the younger node $u$
Set birth$(R)$ ← birth(root$(u)$), death$(R)$ = $\alpha$, Core$(R)$ ←
  Live(root$(u)$), bar$(w)$ ← ind$(R)$ for each $w \in$ Live(root$(u)$)
if bar$(v)$ is defined then sup$(R)$ = bar$(v)$
else  heir$(R)$ = $v$ (the region of $v$ will enter Map later)
end if
if height(root$(u)$) ≤ height(root$(v)$) then
    uplink(root$(u)$) ← root$(v)$
    height(root$(v)$) + + if height(root$(u)$) = height(root$(v)$)
else  uplink(root$(v)$) ← root$(u)$
    Live(root$(u)$) ← Live(root$(v)$)
    birth(root$(u)$) ← birth(root$(v)$)
end if
```

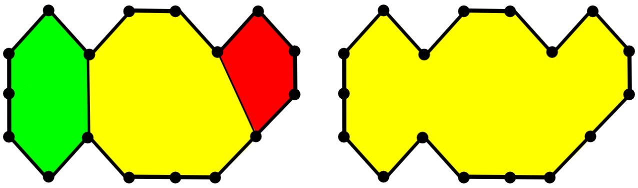

Fig. 9. Initial segmentation: 3 regions. Final segmentation: 1 region.

The k-th widest gap in persistence. We find the k-th gap between decreasing persistences of regions in sorted Map(α) in time $O(n \log n)$. In the conditions of Theorem 9, the 1st gap separates m dots (birth, death) corresponding to true cycles of a graph G. Even if the conditions do not hold, the widest gap gives an approximation to the expected number m of regions, which can also be set by a user, for a final segmentation below.

Indices sup(R) **of superior regions.** We go through each region $R \in$ Map(α) and build the 1-1 correspondence old index $\mapsto$ new index in sorted Map(α). We go again through each R and access the node heir(R) whose bar index bar(heir(R)) is the original (non-sorted) index of the *superior* region that merged with R. Using the 1-1 correspondence of indices in Map(α) above, we know the new index sup(R) of this *superior* or more persistent region with a higher persistence that merged with R.

Final segmentation into m regions. Now we form m regions, where m can be the number of dots in the 1st (or k-th) subdiagram of PD$\{C^\alpha\}$ or m can be user-defined. We go through all regions R of sorted Map(α) starting from the least persistent region. If the current index ind(R) $> m$, we add the list Core(R) to Core(sup(R)), which enlarges the superior region that has the index sup(R) and a higher persistence. If ind(sup(R)) $> m$, the region sup(R) will also merge with its superior later. After merging all regions of ind(R) $> m$ with their superiors one by one, we output lists Core(R) of triangles for ind(R) $= 1, \dots, m$.

5. Fast running time and reconstruction guarantees

The main results are Theorem 7, guarantees for boundaries in Theorem 9 and global stability of output in Corollary 10.

Theorem 7 (fast computation of persistent contours). *For any point cloud C of n points in the plane, the algorithm in section 4 computing the full hierarchy of segmentations has the time complexity $O(n \log n)$ and memory space $O(n)$.*

Proof of Theorem 7. A Delaunay triangulation Del(C) for a cloud $C \subset \mathbb{R}^2$ of n points has $l = O(n)$ triangles and is found in time $O(n \log n)$ (de Berg et al., 2008, section 9.1). The 'while' loop in Algorithm 1 goes once through not more than $O(n)$ edges in Del(C). We can associate to each edge e its two incident triangles $T_u, T_v \subset$ Del(C) in advance, so identifying the corresponding nodes u, v in line 8 is easy. We prove that finding root(u), root(v) in line 9 by going along uplinks in any tree of Forest(α) with l nodes requires $O(\log l) = O(\log n)$ steps.

The height of a tree can increase only in Case 2 (from 1 to 2) or in Case 3, where the height jumps by 1 after we link two trees of the same height. In Case 3 we always link a shorter tree to a taller one. So any two paths in a tree from a root to terminal nodes (leaves) can differ by at most 1, where we include all trivial paths consisting of a single node.

Hence almost any node is linked to at least nodes one level down, except all terminal nodes and some nodes only one level up. Then any tree of height $h \geq 1$ should contain at least 2^{j-1} nodes at level $1 \leq j \leq h-1$ plus at least one node at level h, so at least $1 + 2 + 2^2 + \dots + 2^{h-2} + 1 = 2^{h-1}$ nodes in total. If $l \geq 2^{h-1}$, then the height is $h \leq 1 + \log_2 l = O(\log l)$.

All other steps in the 'while' loop from section 4 need $O(1)$ time. Then we spend $O(n \log n)$ time for sorting $O(n)$ entries in Map(α). The lists Live(v) of triangles are disjoint in Forest(α) as well as similar lists Core(R) in Map(α). Then $O(n)$ Delaunay edges, triangles or corresponding nodes with attributes need only $O(n)$ space in Forest(α) and similarly in Map(α). □

Continuous maps between spaces $f_0, f_1 : X \to Y$ are called *homotopic* if they can be included into a continuous family of maps $f_t : X \to Y$, $t \in [0, 1]$. A set $S \subset \mathbb{R}^2$ is *contractible* to a point $q \in S$ if the identity id : $S \to S$ is homotopic to $S \to q$.

Let L be any closed non-self-intersecting loop in $\mathbb{R}^2$. We consider all offsets L^α when α is increasing. If $\alpha > 0$ is small, L^α is the thickening of L with the radius α. There is a first critical scale α when the internal boundary of L^α touches itself, so L^α is no longer a topological annulus. There is a last critical scale α when the internal boundary of L^α shrinks to a point, so L^α becomes contractible, which is true for all larger radii α.

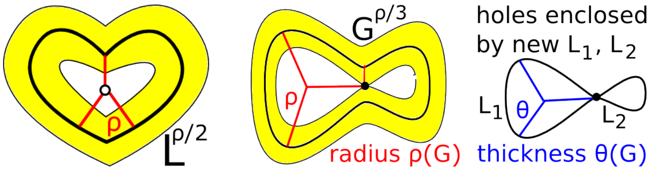

Fig. 10. The 'heart' graph has $\theta = 0$. The 'figure-eight' graph has $\theta > 0$.

A round disk is contractible to its center, but we can not contract a circle to one point always staying within the circle. Below we consider closed loops that go along boundaries of regions in $\mathbb{R}^2 - G$. The internal loop of the graph ⦶ goes along the short vertical edge twice, namely up and down.

Definition 8 (radius ρ of a cycle, thickness θ of a graph G). *A cycle L of a graph $G \subset \mathbb{R}^2$ is* basic *if L is non-self-intersecting and encloses a bounded region of $\mathbb{R}^2 - G$. When α is increasing, the hole enclosed by the α-offset L^α is born at $\alpha = 0$ and dies at the scale $\alpha = \rho(L)$ that is called the* radius *of the cycle L. So the hole enclosed by L has the life span $0 \leq \alpha < \rho(L)$.*

In general, when α is increasing, new holes can be born in G^α, let they be enclosed by $L_1, \dots, L_k$ at their birth times. The thickness $\theta(G) = \max\limits_{j=1,\dots,k} \rho(L_j)$ *is the maximum persistence of these smaller holes born during the evolution of offsets G^α. If no such holes appear, then $\theta = 0$, otherwise $\theta > 0$, see Fig. 10.*

If a cycle $L \subset \mathbb{R}^2$ encloses a convex region, then the only hole of L^α completely dies when α is the radius $\rho(L)$, so $\theta(L) = 0$.

The heart-shaped cycle L in the first picture of Fig. 10 encloses a non-convex region, however no new holes are born in

L^α, so $\theta(L) = 0$. The figure-eight-shaped graph G in the second picture of Fig. 10 has a positive thickness equal to the radius $\rho(L_1)$ of the largest cycle born in G^α when α is increasing.

A cloud C is an *ε-sample* of a graph $G \subset \mathbb{R}^2$ if $C \subset G^\varepsilon$ and $G \subset C^\varepsilon$. Theorem 9 gives conditions on C and G when the 1st segmentation in the output hierarchy for C has boundary contours close to all basic cycles of the unknown graph G.

Theorem 9 (guarantees for the 1st segmentation). *Let C be any ε-sample of a connected graph $G \subset \mathbb{R}^2$ with a thickness $\theta(G) \geq 0$ and $m \geq 1$ basic cycles having ordered radii $\rho_1 \leq \ldots \leq \rho_m$. If $\rho_1 > 7\varepsilon + \theta(G) + \max\limits_{i=1,\ldots,m-1}\{\rho_{i+1} - \rho_i\}$, then the 1st segmentation of C from Definition 4 has exactly m regions whose boundary contours are in the 2ε-offset $G^{2\varepsilon} \subset \mathbb{R}^2$.*

The cloud C in Fig. 2 can be considered as an ε-sample of the graph $G = C(\sqrt{2})$ in Fig. 3 for $\varepsilon = \sqrt{2}$. Then $\theta(G) = R_2 - 2 = 0.5$ is the maximum persistence of a new hole born in G^α. Since G has only one radius $\rho_1 = R_1 \approx 3.642$, which is the minimum scale when G^α becomes contractible, the inequality of Theorem 9 fails. However PD$\{C^\alpha\}$ in Fig. 6 above the widest gap has 1 dot corresponding to one region enclosed by G in Fig. 9. So the algorithm may give a correct reconstruction beyond the guarantees of Theorem 9 when a noisy sample is ‘uniform’.

Since resulting contours pass through points of C, the output is locally sensitive to perturbations of C. However, the result below confirms a global stability of boundaries in a small offset.

Corollary 10. *In the conditions of Theorem 9 if another cloud $\tilde{C}$ is δ-close to C, then the boundaries in the 1st segmentation from Definition 4 for the perturbed cloud $\tilde{C}$ are within the $(2\delta + 4\varepsilon)$-offset of the boundaries in the 1st segmentation for C.*

Theorem 9 and Corollary 10 are proved in (Kurlin, 2015, Appendix B). The key tool is the stability of persistence saying that the diagram PD$\{C^\alpha\}$ is in the ε-neighborhood of PD$\{G^\alpha\}$. The inequality in Theorem 9 implies that the diagonal gap $\{0 < y - x < \rho_1\}$ is widest in PD$\{G^\alpha\}$ and contains a widest gap of the perturbed diagram PD$\{C^\alpha\}$. Hence, using only C without extra parameters, we can find the widest gap in PD$\{C^\alpha\}$ separating persistent contours form all noise near the diagonal.

6. Conclusions and experiments on synthetic and real data

Clouds C of $n = 1000$ points in Fig. 11–12 were randomly sampled around graphs $G \subset \mathbb{R}^2$ (the regular octagon with 4 big diagonals in Fig. 11, a lattice graph in Fig. 12) as follows:

(1) choose a random seed point p in the graph G so that the distribution of seeds over the total length of G is uniform;

(2) choose a final point q in $[p-\varepsilon, p+\varepsilon] \times [p-\varepsilon, p+\varepsilon]$, which is the square neighborhood of a size ε around the seed p.

The noise bound ε is needed only for generating a cloud C, the algorithm uses only C, not ε. The 2nd picture of Fig. 11 is the 1D persistence diagram PD$\{C^\alpha\}$ with exactly 8 dots above the yellow widest gap. The initial regions in the 3rd picture of Fig. 11 are in a 1-1 correspondence with all dots from PD$\{C^\alpha\}$.

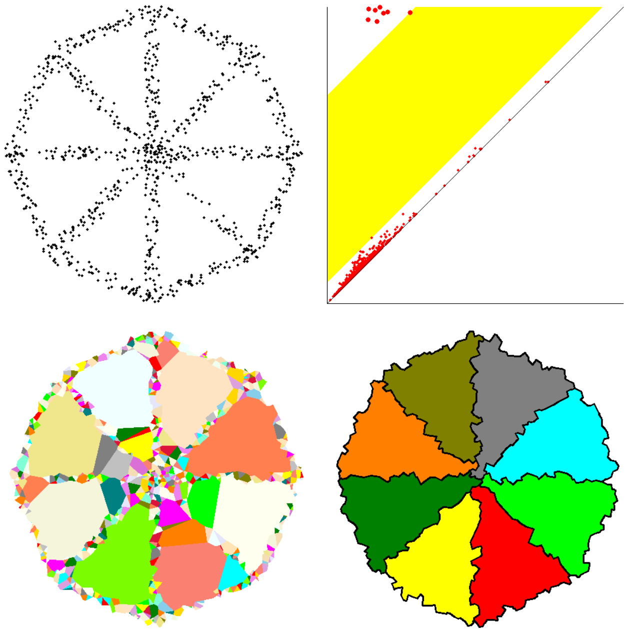

Fig. 11. Top: cloud C and 1D persistence diagram PD$\{C^\alpha\}$ with yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

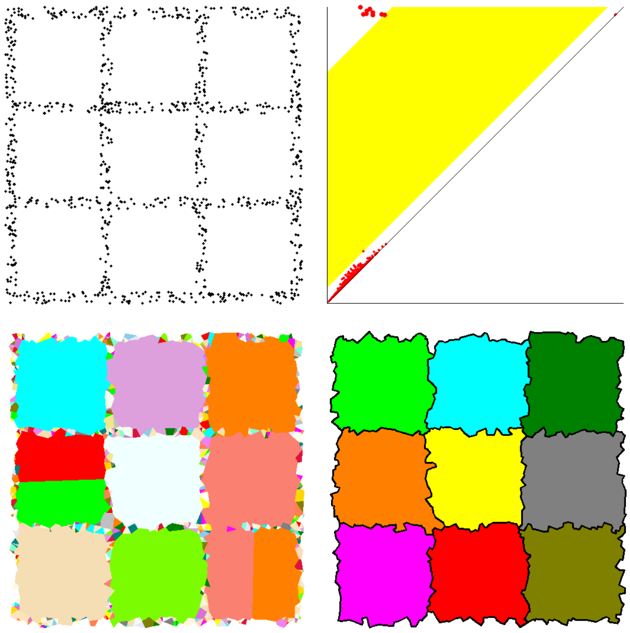

Fig. 12. Top: cloud C and 1D persistence diagram PD$\{C^\alpha\}$ with yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

The final segmentation in the 4th picture has only 8 regions obtained by merging ‘noisy’ regions of lower persistence.

The initial segmentation in the 3rd picture of Fig. 12 contains more than 9 large regions. However only 9 regions have a large enough persistence, all others correctly merge them in the 4th picture. So the size doesn’t matter, but the persistence does!

The clouds C in Fig. 13–14 are samples of binary images. C is rather sparse in Fig. 13, but the widest gap in PD$\{C^\alpha\}$ separates 2 dots corresponding to the final 2 non-convex regions.

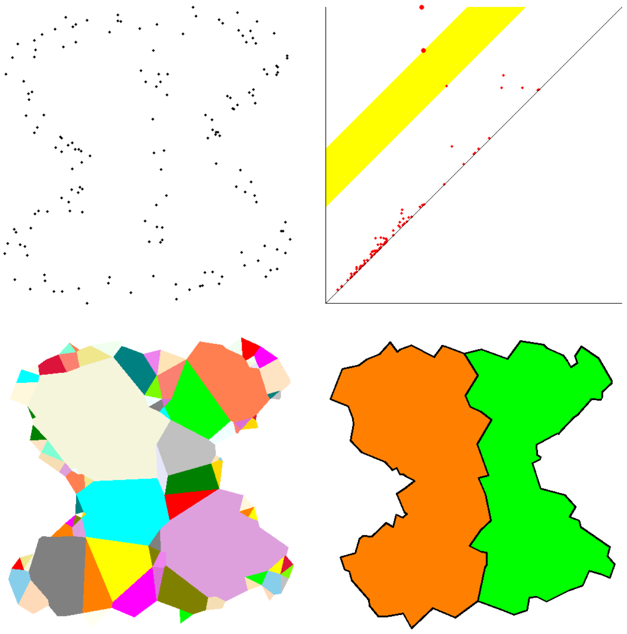

Fig. 13. Top: cloud C of $n = 159$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

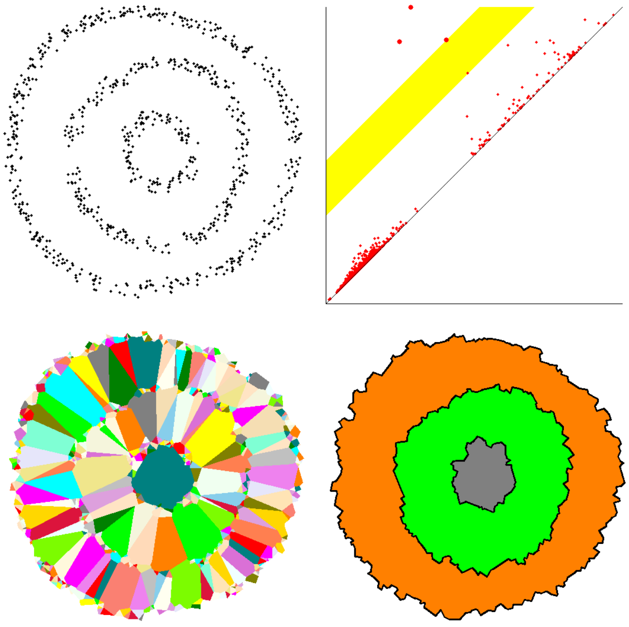

Fig. 14. Top: cloud C of $n = 852$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

The 1st pictures in Fig. 15–18 are from the Berkely segmentation database, see Martin et al. (2001). The 2nd pictures are clouds C obtained as Canny edge points with the low threshold 75 and ratio 3. The clouds C in Fig. 15 and 18 have a lot of clutter, but the final segmentation in the 4th pictures correctly identify a shape of the object corresponding to the only dot above the 1st widest gap in the diagrams from the 3rd pictures.

The final 2 pictures in Fig. 16 and 18 show the 2nd segmentations from the hierarchy, which separate two hands from the face in Fig. 16 and the boat from its mirror image in Fig. 18. More experiments are in (Kurlin, 2015, Appendix C).

The summary of the key contributions is below.

- The $O(n \log n)$ time algorithm from section 4 accepts any cloud of points in $\mathbb{R}^2$ without extra parameters and outputs a hierarchy of segmentations selected by their persistence.
- Theorem 9 proves that the persistent contours approximate true contours of a graph G given only by a noisy sample C.
- Corollary 10 guarantees a global stability of boundaries in the 1st segmentation of C under perturbations of a given cloud C.

The C++ code is at http://kurlin.org/projects/cloud-analysis.cpp. In comparison with the earlier conference version in Kurlin (2014a), Theorem 9 has been extended from the partial case $\theta(G) = 0$ to a wider class of graphs with any thickness $\theta(G) \geq 0$. Since the algorithm needs no parameters, there is a lot of freedom to preprocess real data and extract a cloud satisfying Theorem 9. Here are further open problems.

- Smooth boundary contours in a final segmentation of C.
- Extend Theorem 9 to noisy samples with unbounded noise.
- Use persistence to locate holes in high-dimensional clouds.

The author is open to collaboration on any related projects and thanks all reviewers in advance for helpful suggestions.

References

Attali, D., Glisse, M., Hornus, S., Lazarus, F., Morozov, D., 2009. Persistence-sensitive simplification of functions on surfaces in linear time, in: TopoInVis: Topology-based Methods in Visualization, Springer.

de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. Computational Geometry: Algorithms and Applications. Springer.

Chen, C., Freedman, D., Lampert, C., 2011. Enforcing topological constraints in random field image segmentation, in: Proceedings of CVPR: Computer Vision and Pattern Recognition, pp. 2089–2096.

Chernov, A., Kurlin, V., 2013. Reconstructing persistent graph structures from noisy images. Image-A 3, 19–22.

Cohen-Steiner, D., Edelsbrunner, H., Harer, J., 2007. Stability of persistence diagrams. Discrete and Computational Geometry 37, 103–130.

Edelsbrunner, H., 1995. The union of balls and its dual shape. Discrete and Computational Geometry 13, 415–440.

Edelsbrunner, H., Harer, J., 2010. Computational topology. An introduction. AMS, Providence.

Kurlin, V., 2014a. Auto-completion of contours in sketches, maps and sparse 2d images based on topological persistence (conference version of the current paper), in: Proceedings of CTIC: Computational Topology in Image Context, pp. 594–601. URL: `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7034735`.

Kurlin, V., 2014b. A fast and robust algorithm to count topologically persistent holes in noisy clouds, in: Proceedings of CVPR: Computer Vision and Pattern Recognition, pp. 1458–1463.

Kurlin, V., 2015. A fast persistence-based segmentation of noisy 2d clouds with provable guarantees (full version of the current paper) URL: `http://kurlin.org/projects/cloud2D-segmentation-full.pdf`.

Letscher, D., Fritts, J., 2007. Image segmentation using topological persistence, in: Proceedings of CAIP: Computer Analysis of Images and Patterns, pp. 587–595.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. ICCV, pp. 416–423.

Saund, E., 2003. Finding perceptually closed paths in sketches and drawings. Transactions Pattern Analysis and Machine Intelligence 25, 475–490.

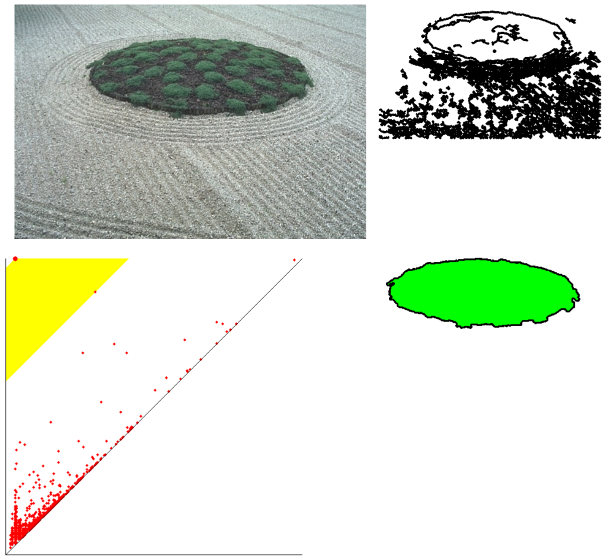

Fig. 15. Top: image 86016 from BSD500 and cloud C of 19627 Canny edge points. Bottom: PD$\{C^\alpha\}$ and final segmentation with a correct object.

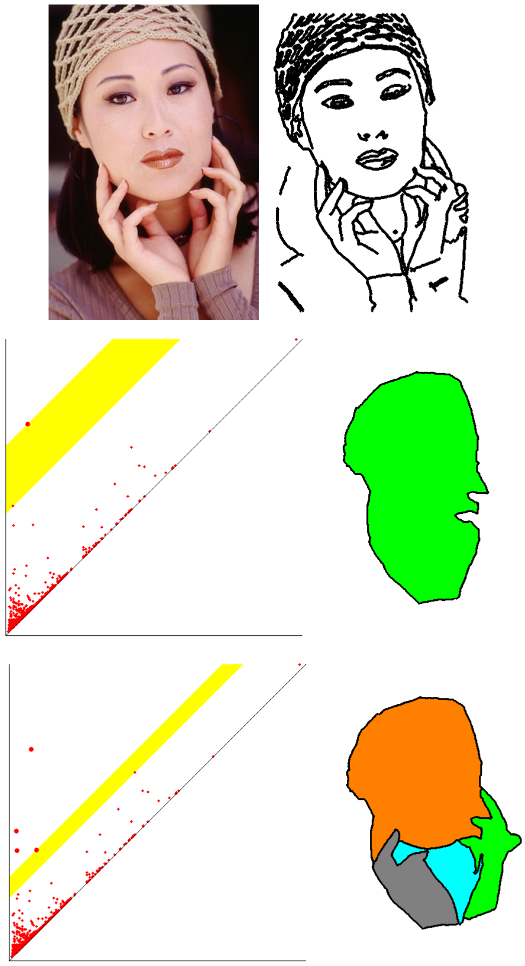

Fig. 16. Top: image 302003 from BSD500 and cloud C of 5388 Canny edge points. Middle: PD$\{C^\alpha\}$ with 1st widest gap and segmentation with 1 region Bottom: PD$\{C^\alpha\}$ with 2nd widest gap and segmentation with 4 regions.

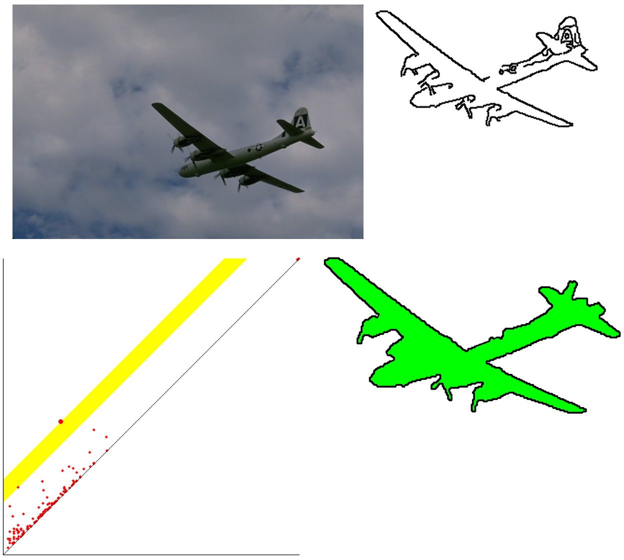

Fig. 17. Top: image 3096 from BSD500 and cloud C of 1614 Canny edge points. Bottom: PD$\{C^\alpha\}$ and final segmentations with a correct object.

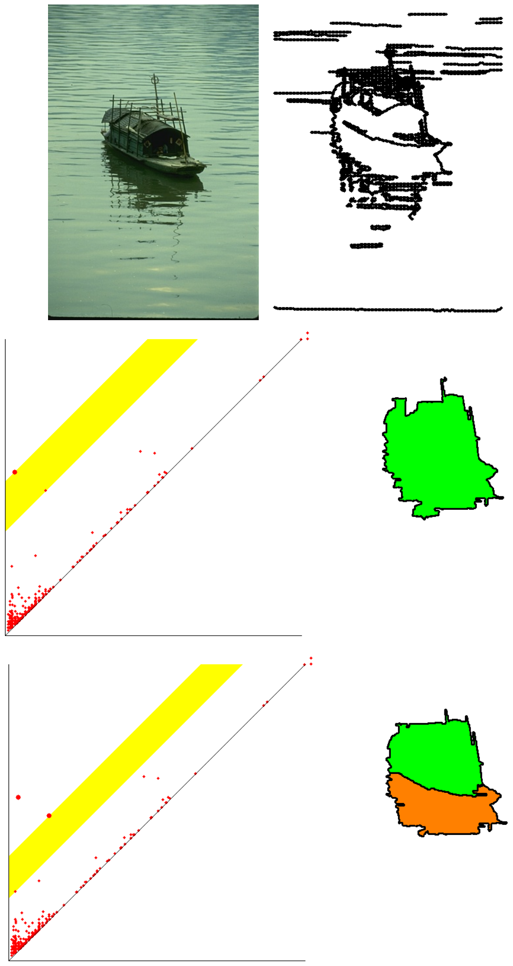

Fig. 18. Top: image 15088 from BSD500 and cloud C of 6565 Canny edge points. Middle: PD$\{C^\alpha\}$ with 1st widest gap and segmentation with 1 region Bottom: PD$\{C^\alpha\}$ with 2nd widest gap and segmentation with 2 regions.

Pattern Recognition Letters
journal homepage: www.elsevier.com

Supplementary materials (appendices with proofs and more experiments) for the paper A fast persistence-based segmentation of noisy 2D clouds with provable guarantees

Vitaliy Kurlin[a,**]

[a]Microsoft Research, 21 Station Road, Cambridge CB1 2FB and Mathematical Sciences, Durham University, Durham DH1 3LE, UK.

ABSTRACT

To avoid any confusion we continue numbering definitions and figures as in the 10-page paper.

Appendix A: α-complexes and their persistent homology

We briefly remind key concepts and results also introducing the α-complexes in a different way to give a new perspective.

Definition 11 (a plane graph, its cycles and holes). *A* plane graph *is a subset $G \subset \mathbb{R}^2$ consisting of finitely many vertices and non-intersecting arcs joining vertices. A* cycle *of G is a subset $L \subset G$ consisting of edges connecting adjacent vertices: p_1 to p_2, p_2 to p_3 and so on until p_k to p_1. A cycle is a closed loop, but may have self-intersections. A cycle $L \subset G$ is called* basic *if L encloses a* hole*, which is a connected region $\mathbb{R}^2 - G$.*

If every bounded region in the complement $\mathbb{R}^2 - G$ of a plane graph is a triangle, then the graph G defines a triangulation on its vertices. The following Delaunay triangulation $\mathrm{Del}(C)$ is a small and quickly computable structure on a cloud $C \subset \mathbb{R}^2$.

Definition 12 (Delaunay triangulation). *For a cloud $C = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ of n points, a* Delaunay triangulation $\mathrm{Del}(C)$ *has all triangles with vertices $p_i, p_j, p_k \in C$ whose circumcircle doesn't enclose any other points of C, see Fig. 2.*

A Delaunay triangulation is not unique if C contains 4 points on the same circle. The boundary edges of $\mathrm{Del}(C)$ form the convex $\mathrm{hull}(C)$ of C. The complement $\mathbb{R}^2 - \mathrm{hull}(C)$ will be called the external region. If $\mathrm{Del}(C)$ has k triangles and b boundary edges, then counting E edges over k triangles gives $3k+b = 2E$. By the Euler formula $n - E + (k + 1) = 2$ in the plane, we conclude that $k = 2n - b - 2$, $E = 3n - b - 3$, so $\mathrm{Del}(C)$ has $O(n)$ edges and triangles. Also $\mathrm{Del}(C)$ is found in time $O(n \log n)$ with $O(n)$ space, see (de Berg et al., 2008, section 9.1).

A Delaunay triangulation $\mathrm{Del}(C)$ is an example of a general 2-dimensional *complex* consisting of vertices, edges and triangles in $\mathbb{R}^2$. To study the shape of a cloud C at different scales, we shall define subcomplexes that contain the elements of $\mathrm{Del}(C)$ whose sizes are bounded above by a fixed radius α.

For a point $p_i \in C$, the *Voronoi cell* consists of all points $q \in \mathbb{R}^2$ that are closer to p_i than to all other points of C, so $V(p_i) = \{q \in \mathbb{R}^2 : d(p_i, q) \le d(p_j, q)$ for any $j \neq i\}$. Then a *Delaunay triangulation* $\mathrm{Del}(C)$ consists of all triangles with vertices $p, q, r \in C$ such that $V(p) \cap V(q) \cap V(r)$ is not empty. If the Voronoi cells are restricted to a scale $\alpha > 0$, we get the α-complexes $C(\alpha)$. For any $p \in \mathbb{R}^2$ and $\alpha > 0$, denote by $B(p; \alpha)$ the closed disk with the center p and radius α.

Definition 13 (α-complexes). *For a finite cloud $C \subset \mathbb{R}^2$, the* α-complex $C(\alpha) \subset \mathbb{R}^2$ *contains all edges between points $p, q \in C$ such that $V(p) \cap B(p; \alpha)$ meets $V(q) \cap B(q; \alpha)$, see (Edelsbrunner and Harer, 2010, section III.4). Similarly, the α-complex $C(\alpha)$ contains all triangles with vertices p, q, r such that the full intersection $V(p) \cap B(p; \alpha) \cap V(q) \cap B(q; \alpha) \cap V(r) \cap B(r; \alpha) \neq \emptyset$.*

The following lemma motivates the concept of the α-complex $C(\alpha)$, which is a simpler object than the α-offset C^α.

Lemma 14. *Edelsbrunner (1995) For any scale α, the α-offset C^α of a cloud $C \subset \mathbb{R}^2$ continuously deforms to $C(\alpha) \subset \mathbb{R}^2$.*

Definition 15 (homology group H_1 of a complex S). *Cycles of a complex S can be algebraically written as linear combinations of edges with coefficients 0 or 1 in the group $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$. The vector space C_1 consists of all these linear combinations. The boundaries of all triangles in S (as cycles of 3 edges) generate the subspace $B_1 \subset C_1$. The quotient space C_1/B_1 is the* 1-dimensional homology *group $H_1(S)$.*

[**] Corresponding author: Tel.: +44-1913343081; fax: +44-1913343051; *e-mail:* vitaliy.kurlin@gmail.com (Vitaliy Kurlin)

Definition 16 (the bottleneck distance). *Let* $\|p - q\|_\infty = \max\{|x_1 - x_2|, |y_1 - y_2|\}$ *be the distance between* $p = (x_1, y_1)$, $q = (x_2, y_2)$ *in* $\mathbb{R}^2$. *The* bottleneck distance *between persistence diagrams* PD *and* PD′ *is* $d_B = \inf_\psi \sup_{q\in \mathrm{PD}} \|q - \psi(q)\|_\infty$ *over all bijections* $\psi : \mathrm{PD} \to \mathrm{PD}'$ *of multi-sets* PD *and* PD′.

Definition 17 (ε-sample). *For any* $\varepsilon > 0$, *a finite cloud* $C \subset \mathbb{R}^2$ *is an* ε*-sample of a graph* $G \subset \mathbb{R}^2$ *if* $C \subset G^\varepsilon$ *and* $G \subset C^\varepsilon$.

Stability Theorem 18 below is a key foundation of topological data analysis saying that the persistence diagram is stable under perturbations of original data. We quote only a simple version of the Stability Theorem for filtrations of offsets.

Theorem 18 (stability). *Cohen-Steiner et al. (2007)*
If a finite cloud $C \subset \mathbb{R}^2$ *of points is an* ε*-sample of a plane graph* $G \subset \mathbb{R}^2$, *then we have* $d_B(\mathrm{PD}\{G^\alpha\}, \mathrm{PD}\{C^\alpha\}) \leq \varepsilon$.

By Lemma 14 any offset C^α has the same homology group H_1 as $C(\alpha)$. So we may replace the filtration $\{C^\alpha\}$ of offsets by the filtration$\{C(\alpha)\}$ of simpler complexes in Theorem 18.

Appendix B: detailed proofs of Theorem 9 and Corollary 10

Proof of Theorem 9. The homology $H_1(G)$ is generated by m basic cycles $L_1, \dots, L_m$ that enclose m holes (bounded regions in the complement $\mathbb{R}^2 - G$). These m cycles give dots $(0, \rho_i)$ in the vertical axis of the 1D persistence diagram $\mathrm{PD}\{G^\alpha\}$.

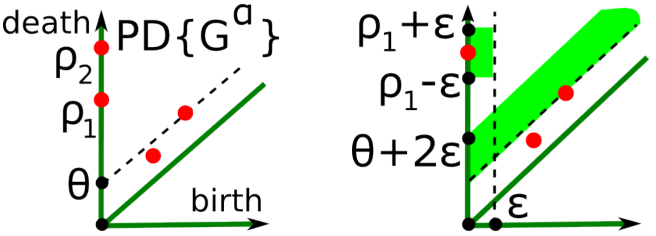

Fig. 19. Left: 1D persistence diagram $\mathrm{PD}\{G^\alpha\}$ **for a graph** G **satisfying Theorem 9. Right: perturbed diagram** $\mathrm{PD}\{C^\alpha\}$ **for a noisy** ε**-sample** C **of** G.

All other dots in the 1D persistence diagram $\mathrm{PD}\{G^\alpha\}$ come from smaller holes in the α-offsets G^α that were born later. The maximum persistence death − birth of these holes is bounded above by the thickness $\theta(G)$, see Definition 8 and Fig. 19.

The given inequality $\rho_1 > 7\varepsilon + \theta(G) + \max\limits_{i=1,\dots,m-1}\{\rho_{i+1} - \rho_i\}$ guarantees that the widest diagonal gap $\{\theta(G) < y - x < \rho_1\}$ in $\mathrm{PD}\{G^\alpha\}$ is wider than any other gaps including the higher gaps $\rho_{i+1} - \rho_i$ between the dots $(0, \rho_i) \in \mathrm{PD}\{G^\alpha\}$, $i = 1, \dots, m - 1$.

By Stability Theorem 18 the perturbed diagram $\mathrm{PD}\{C(\alpha)\}$ is in the ε-offset of $\mathrm{PD}\{G^\alpha\} \subset \cup_{i=1}^m (0, \rho_i) \cup \{y - x < \theta(G)\}$ with respect to the L_∞ metric on $\mathbb{R}^2$. All noisy dots near the diagonal in $\mathrm{PD}\{C(\alpha)\}$ can not be higher than $\theta(G) + 2\varepsilon$ after projecting along the diagonal $\{x = y\}$ to the vertical axis $\{x = 0\}$.

The remaining dots can not be lower than $\rho_1 - 2\varepsilon$ after the same projection $(x, y) \mapsto y - x$. Hence the smaller diagonal strip $\{\theta(G) + 2\varepsilon < y - x < \rho_1 - 2\varepsilon\}$ of the vertical width $\rho_1 - 4\varepsilon - \theta(G)$ is still empty in the perturbed diagram $\mathrm{PD}\{C(\alpha)\}$.

By Stability Theorem 18 any dot $(0, \rho_i) \in \mathrm{PD}\{G^\alpha\}$, $i \geq 2$, can not jump lower than the line $y - x = \rho_i - 2\varepsilon$ or higher than $y - x = \rho_i + \varepsilon$. Then the widest diagonal gap between these perturbed dots has a vertical width at most $\max\limits_{i=1,\dots,m-1}\{\rho_{i+1} - \rho_i\} + 3\varepsilon$.

Since all dots near the diagonal have diagonal gaps not wider than $\theta(G) + 2\varepsilon$, the 2nd widest gap in the perturbed diagram $\mathrm{PD}\{C(\alpha)\}$ always has a vertical width smaller than $\rho_1 - 4\varepsilon - \theta(G)$. Hence the 1st widest gap in $\mathrm{PD}\{C(\alpha)\}$ covers the diagonal strip $\{\theta(G) + 2\varepsilon < y - x < \rho_1 - 2\varepsilon\}$, which is within the 1st widest gap $\{\theta(G) < y - x < \rho_1\}$ in the original diagram $\mathrm{PD}\{G^\alpha\}$.

Then the subdiagram $\mathrm{PD}_1\{C(\alpha)\}$ above the line $y - x = \rho_1 - 2\varepsilon$ contains exactly m perturbations of the original dots $(0, \rho_i)$ in the vertical strip $\{0 \leq x \leq \varepsilon\}$. Hence the boundary contours ∂C from the 1st segmentation of C contain exactly m critical edges corresponding to the m dots in the subdiagram $\mathrm{PD}_1\{C(\alpha)\}$.

It remains to prove that the boundary contours ∂C in the 1st segmentation of a cloud C are 2ε-close to G. Let the *critical* scale $\alpha(C)$ be the maximum birth over all dots in $\mathrm{PD}_1\{C(\alpha)\}$. These dots are at most ε away from their corresponding points $(0, \rho_i)$ in the vertical axis, so critical scale $\alpha(C)$ is at most ε.

A longest edge e of any contour persisting over birth $\leq \alpha <$ death has the half-length equal to its birth, because adding e made the contour closed giving a birth its homology class.

Then all edges in the boundary contours ∂C have half-lengths at most $\alpha(C) \leq \varepsilon$. Hence ∂C is covered by the disks with the radius ε and centers at all points of C, so $\partial C \subset C^\varepsilon \subset G^{2\varepsilon}$. □

Proof of Corollary 10. The condition that the perturbed cloud $\tilde{C}$ is δ-close to the original cloud C, which is ε-close to the graph G, implies that $\tilde{C}$ is $(\delta + \varepsilon)$-close to the graph $G \subset \mathbb{R}^2$.

Theorem 9 for the ε-sample C and $(\delta + \varepsilon)$-sample $\tilde{C}$ says that the boundaries ∂C of the 1st segmentation are 2ε-close to G and similarly the boundaries $\partial\tilde{C}$ are $(2\delta + 2\varepsilon)$-close to the graph G.

Hence these boundaries are $(2\delta + 4\varepsilon)$-close to each other. □

Appendix C: more experiments on synthetic and real data

Fig. 20–27 show that many graphs are correctly reconstructed from noisy samples generated as in Fig. 11–12.

Fig. 28 shows binary images whose random samples are processed in Fig. 13–14 in the paper and in Fig. 29–32 below. Despite the sparse cloud C in Fig. 31 has only 137 points, the widest gap in the persistence diagram clearly separates 3 dots leading to 3 expected regions, see the last picture in Fig. 28.

As in the paper, Canny edge points are extracted from images in Fig. 33–36 with the same low threshold 75 and ratio 3.

The cloud C in the 2nd picture of Fig. 35 has many noisy outliers. However the 1st segmentation gives a single correct shape in the 4th picture obtained without extra input parameters.

Fig. 34 and 36 show 2 segmentations from the hierarchy corresponding to the 1st and 2nd widest gaps in persistence.

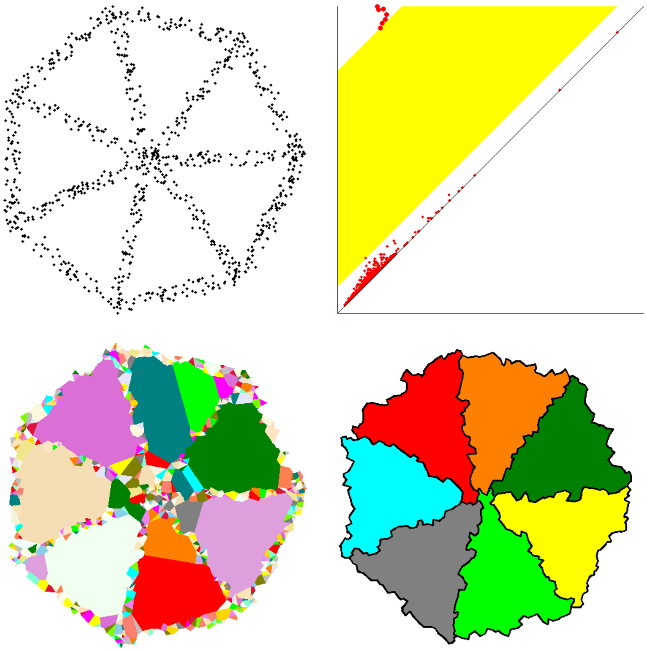

Fig. 20. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^\alpha\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

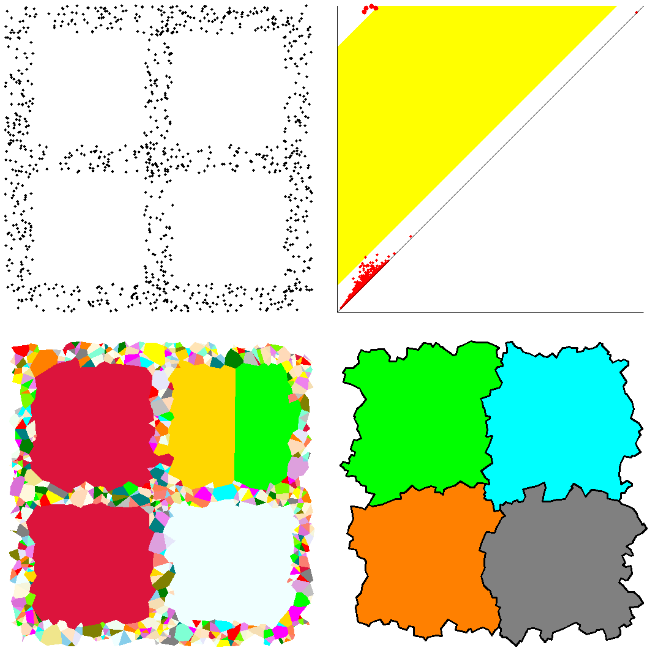

Fig. 21. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^\alpha\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

References

de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. Computational Geometry: Algorithms and Applications. Springer.

Cohen-Steiner, D., Edelsbrunner, H., Harer, J., 2007. Stability of persistence diagrams. Discrete and Computational Geometry 37, 103–130.

Edelsbrunner, H., 1995. The union of balls and its dual shape. Discrete and Computational Geometry 13, 415–440.

Edelsbrunner, H., Harer, J., 2010. Computational topology. An introduction. AMS, Providence.

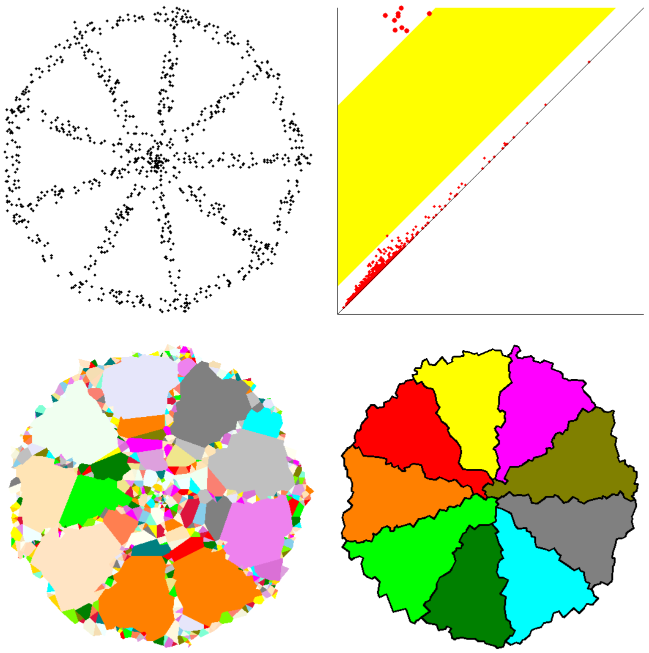

Fig. 22. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^\alpha\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

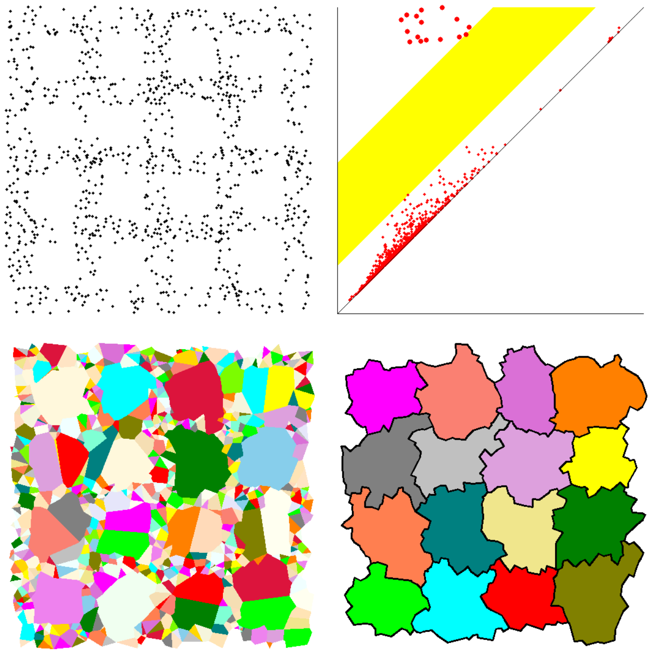

Fig. 23. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^\alpha\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

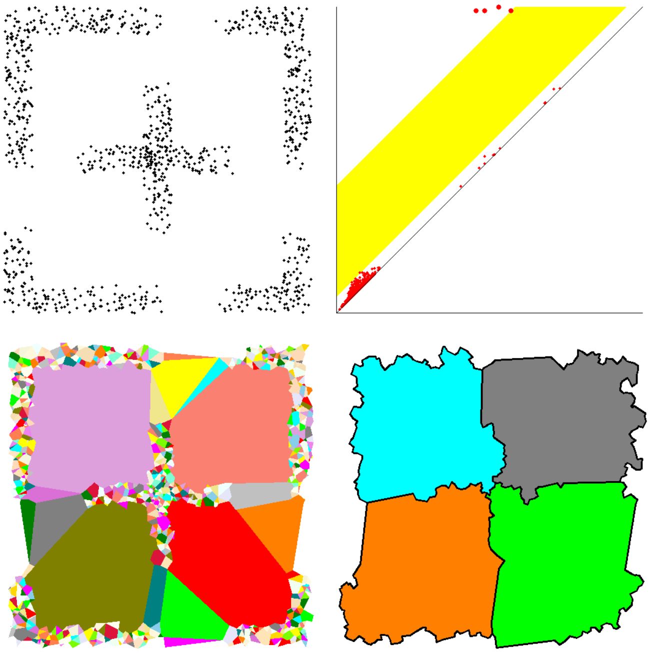

Fig. 24. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

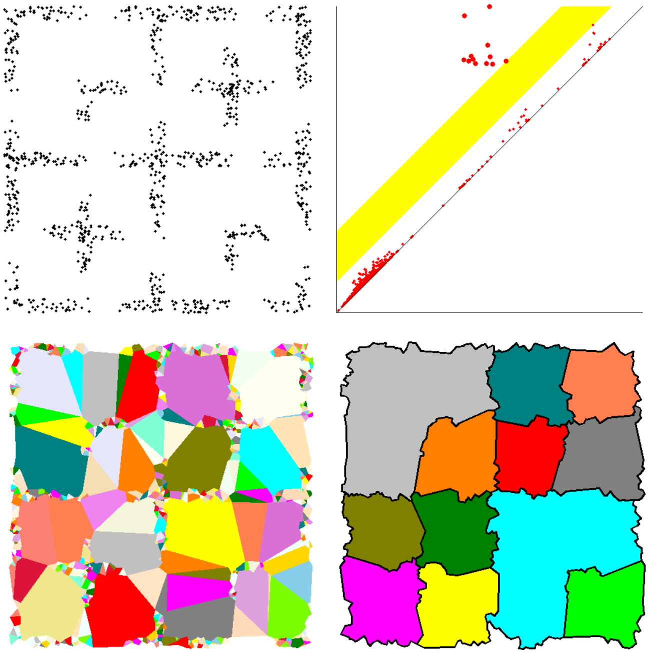

Fig. 25. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

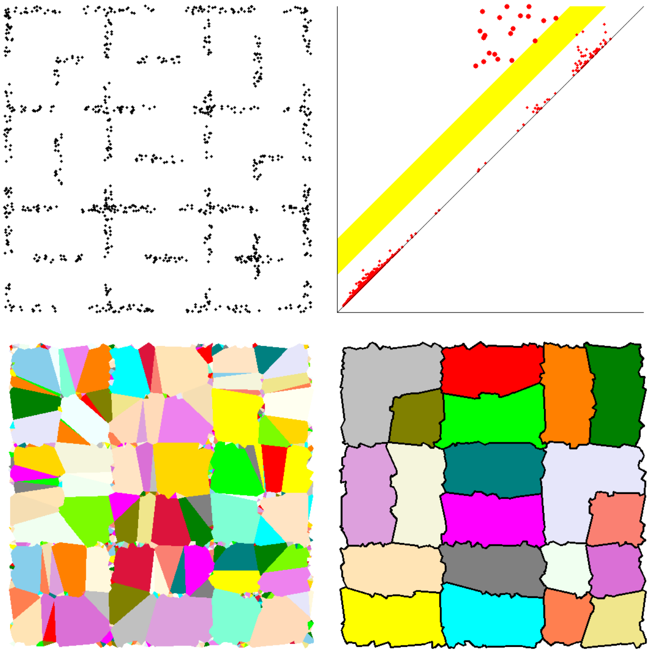

Fig. 26. Top: cloud C of $n = 1000$ points, diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

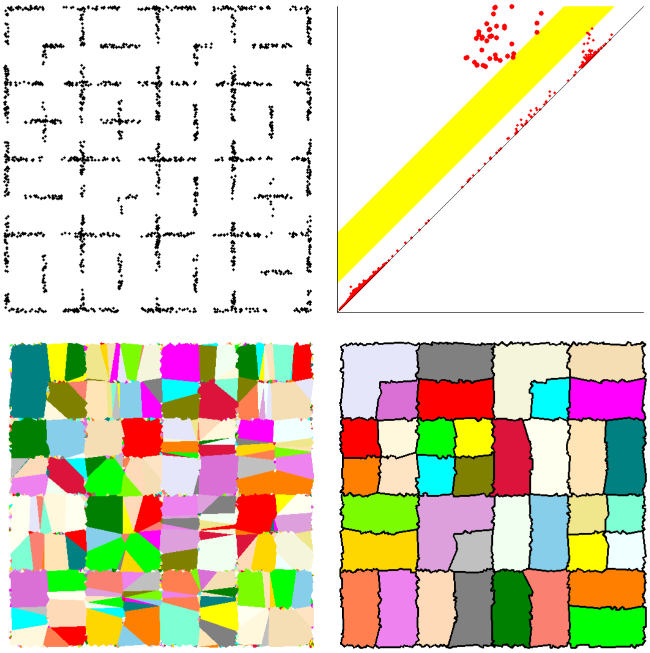

Fig. 27. Top: cloud C of $n = 2000$ points, diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

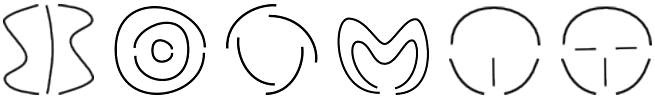

Fig. 28. Binary images whose samples are processed in Fig. 13–14, 29–32.

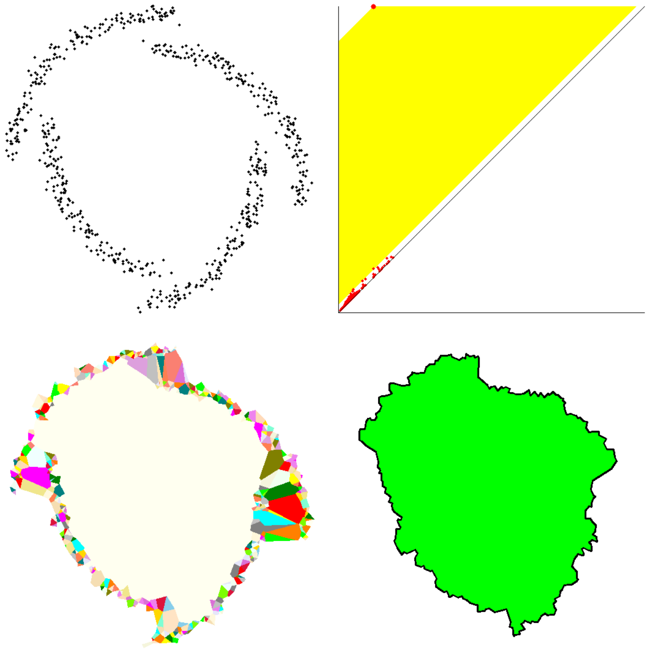

Fig. 29. Top: cloud C of $n = 636$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

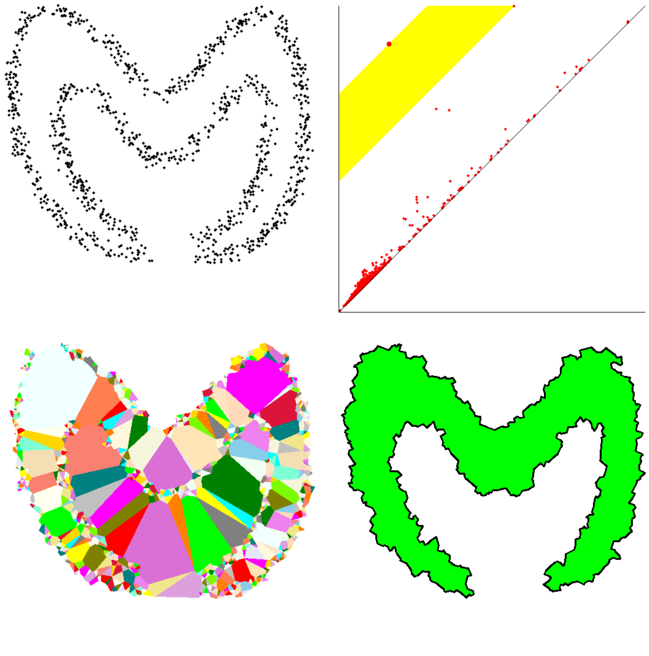

Fig. 30. Top: cloud C of $n = 1146$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the widest gap. Bottom: initial and final segmentations (best viewed in color).

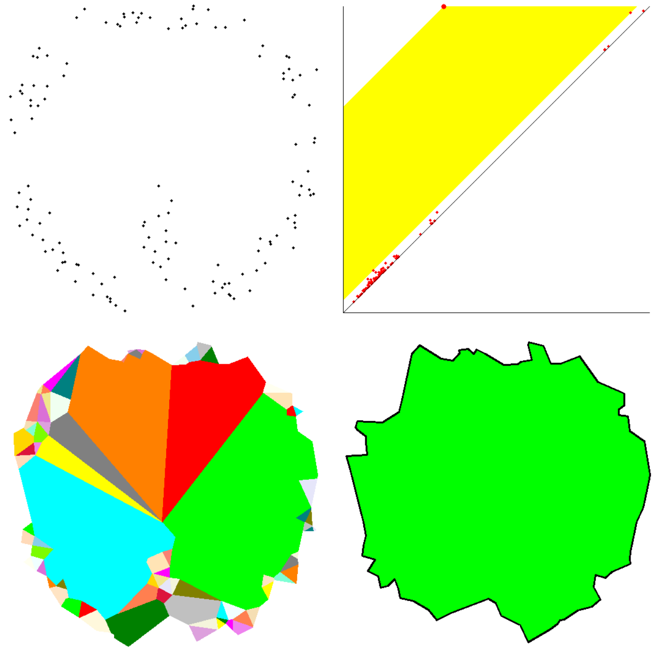

Fig. 31. Top: cloud C of $n = 137$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

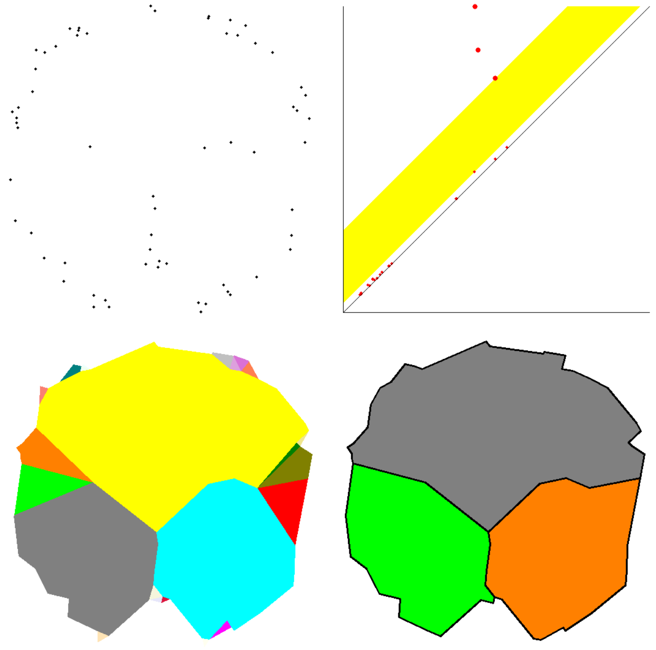

Fig. 32. Top: cloud C of $n = 63$ points and diagram $\mathrm{PD}\{C^{\alpha}\}$ with the yellow widest gap. Bottom: initial and final segmentations (best viewed in color).

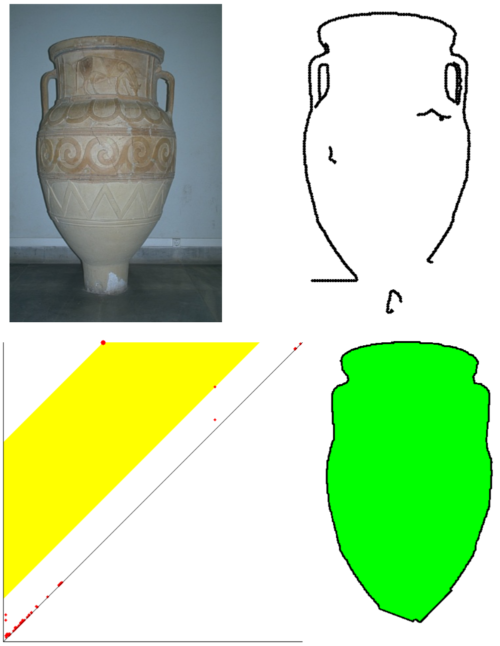

Fig. 33. Top: image 388006 from BSD500 and cloud C of 1489 Canny edge points. Bottom: $\mathrm{PD}\{C^{\alpha}\}$ **and final segmentations (best viewed in color).**

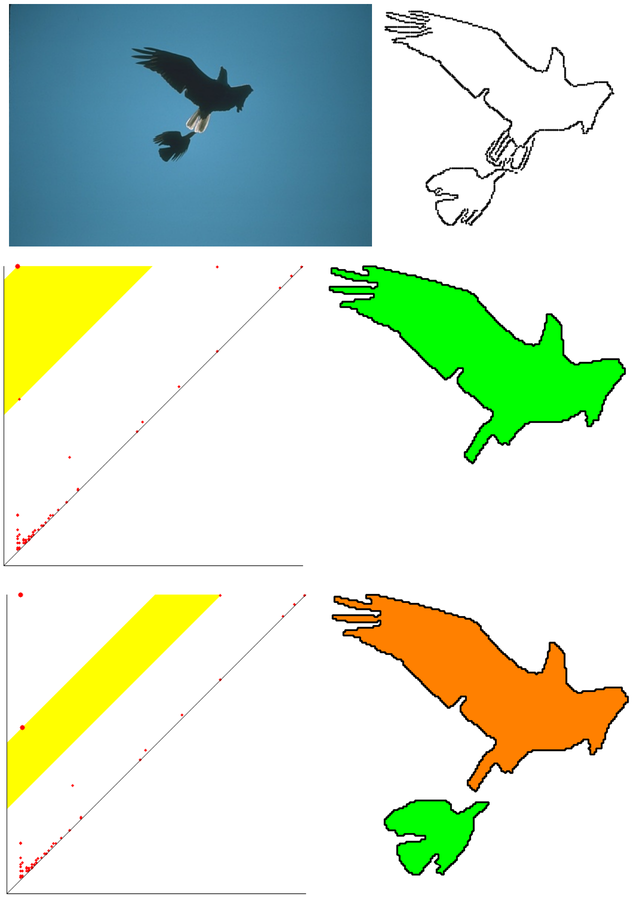

Fig. 34. Top: image 135069 from BSD500 and cloud C of 1136 Canny edge points. Middle: $\mathrm{PD}\{C^{\alpha}\}$ **with 1st widest gap and segmentation with 1 region Bottom:** $\mathrm{PD}\{C^{\alpha}\}$ **with 2nd widest gap and segmentation with 2 regions.**

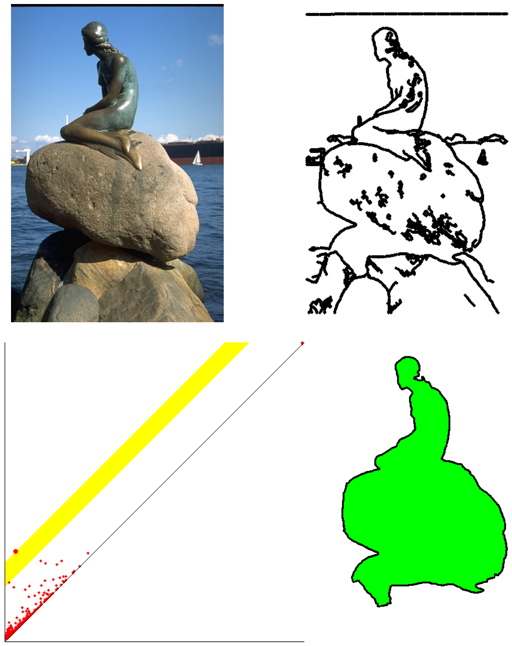

Fig. 35. Top: image 372019 from BSD500 and cloud C of 6637 Canny edge points. Bottom: $\mathrm{PD}\{C^{\alpha}\}$ **and final segmentations (best viewed in color).**

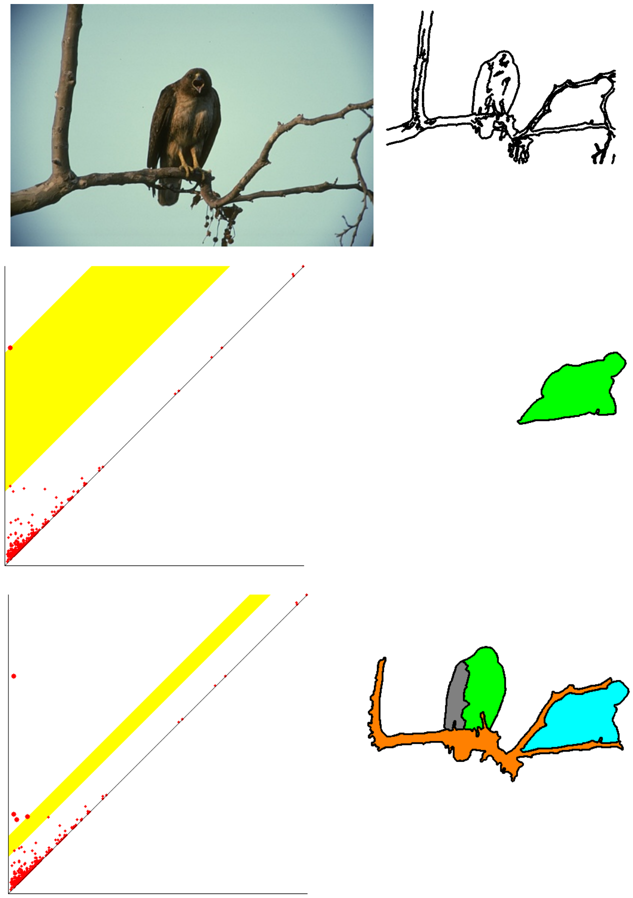

Fig. 36. Top: image 107072 from BSD500 and cloud C of 5394 Canny edge points. Middle: $\mathrm{PD}\{C^{\alpha}\}$ **with 1st widest gap and segmentation with 1 region Bottom:** $\mathrm{PD}\{C^{\alpha}\}$ **with 2nd widest gap and segmentation with 4 regions.**