

CGWeek Young Researchers Forum 2018

Booklet of Abstracts

2018

This volume contains the abstracts of papers at “Computational Geometry: Young Researchers Forum” (CG:YRF), a satellite event of the 34th International Symposium on Computational Geometry, held in Budapest, Hungary on June 11-14, 2018.

The CG:YRF program committee consisted of the following people:

- Don Sheehy (chair), University of Connecticut
- Peyman Afshani, Aarhus University
- Chao Chen, CUNY Queens College
- Elena Khramtcova, Université Libre de Bruxelles
- Irina Kostitsyna, TU Eindhoven
- Jon Lenchner, IBM Research, Africa
- Nabil Mustafa, ESIEE Paris
- Amir Nayyeri, Oregon State University
- Haitao Wang, Utah State University
- Yusu Wang, The Ohio State University

There were 31 papers submitted to CG:YRF. Of these, 28 were accepted with revisions.

Copyrights of the articles in these proceedings are maintained by their respective authors. More information about this conference and about previous and future editions is available online at

<http://www.computational-geometry.org/>

Topology-aware Terrain Simplification

Ulderico Fugacci

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, 8010 Graz, Austria
fugacci@tugraz.at

 <https://orcid.org/0000-0003-3062-997X>

Michael Kerber

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, 8010 Graz, Austria
kerber@tugraz.at

 <https://orcid.org/0000-0002-8030-9299>

Hugo Manet

Département d'Informatique, École Normale Supérieure
45 rue d'Ulm, 75005 Paris, France
hugo.manet@ens.fr

 <https://orcid.org/0000-0003-4649-2584>

Abstract

A common issue in terrain visualization is caused by oversampling of flat regions and of areas with constant slope. For a more compact representation, it is desirable to remove vertices in such areas, maintaining both the topological properties of the terrain and a good quality of the underlying mesh. We propose a simple algorithm that preserves the persistent homology of the height function of the terrain and returns a constrained Delaunay triangulation, where the constraints are defined by edges that cannot be flipped without changing the topology.

2012 ACM Subject Classification Computing methodologies → Mesh models

Keywords and phrases Terrain simplification, topological operators, persistent homology, constrained Delaunay triangulation.

Funding Supported by the Austrian Science Fund (FWF) grant number P 29984-N35.

1 Summary

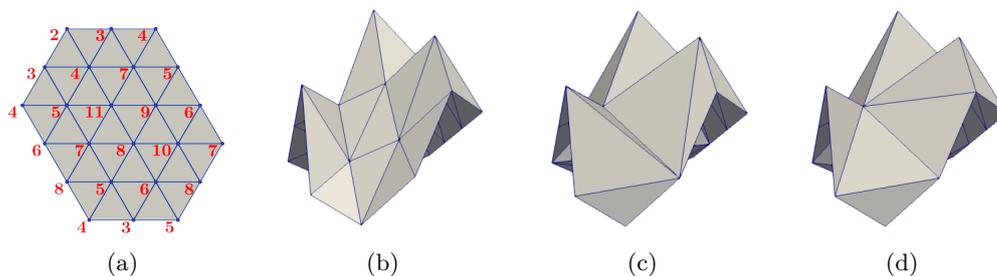


Figure 1 A toy example on which the proposed algorithm is performed.

Let us consider a scalar field $f : P \rightarrow \mathbb{R}$ defined on a finite set of points P in \mathbb{R}^2 and let Σ be the Delaunay triangulation of P (see Figure 1(a) for an example). As shown in Figure 1(b), such a dataset can be represented as a *terrain*, a triangulated surface in \mathbb{R}^3 by

considering, for each $p \in P$, the scalar value $f(p)$ as an altitude attribute. In real datasets, such a representation can be affected by a geometrical oversampling. This is caused by the presence of regions with approximately constant slope represented by a large number of triangles that could be realized through a much more limited number of simplices. In general, it is worthwhile to ask how a terrain can be simplified while maintaining its high-level structure. At the same time, the simplified mesh should be of good quality, e.g., not contain unnecessarily skinny triangles. See [2] for an overview of terrain simplification.

We propose a simplification framework that maintains the persistence diagram defined by the sublevel sets of f on Σ and yields a mesh with certain quality guarantees. Our strategy consists of two phases: first, we remove vertices in the mesh that cause oversampling; done carefully, this preserves the persistent homology, but might deteriorate the mesh quality (Figure 1(c)). In the second phase, we repair the underlying mesh, that is, we improve its quality, again without changing the persistence of the terrain (Figure 1(d)).

Removal phase. We design a criterion to determine if the removal of a vertex v is possible while preserving the persistent homology of the terrain. According to this criterion, verifiable in polynomial time in the number of vertices adjacent to v , a large class of vertices, including for instance any non-critical point v of f whose incident triangles form a convex set, can be removed without affecting the topological properties of the terrain. Then, the focus is on determining which vertices to remove to reduce oversampling. This can be controlled by a heuristic based on the local properties, such as curvature, local density, etc.

Repair phase. After the removals, the main idea is to flip edges in the triangulation to improve its quality. The first question is which edges can be flipped without affecting the persistence of the terrain. Let ab be an edge with the two adjacent triangles abc and abd . Assuming wlog $f(a) < f(b)$, we call ab *feasible* if exactly one value among $f(c)$ and $f(d)$ is contained in the interval $[f(a), f(b)]$.

► **Lemma 1.** *Flipping a feasible edge does not change the persistence diagram of the terrain.*

► **Lemma 2.** *Flipping a feasible edge does not turn a feasible edge infeasible, or vice versa.*

Based on these two lemmas, the repair phase identifies the infeasible edges. Then, it computes the *constrained Delaunay triangulation* [1] of the mesh subject to the infeasible edges as constraints. This triangulation, maximizing the minimal angle among all triangulations containing the infeasible edges, can be obtained from the starting one through a sequence of flips of feasible edges, and hence has the same persistence diagram as the initial terrain.

Outlook. We plan to experimentally evaluate different strategies for vertex removals in the first phase of our framework and carefully compare the results with the state of the art. Moreover, we want to study the variant where the persistence diagram of the simplified terrain is allowed to vary from the original one by a specified amount. For that, an extension of Lemma 1 bounds the change of persistence when flipping an infeasible edge, allowing for a greedy simplification strategy. However, finding the optimal quality triangulation whose persistence is close to the original one appears to be a more difficult problem.

References

- 1 Siu-Wing Cheng, Tamal K Dey, and Jonathan Shewchuk. *Delaunay mesh generation*. CRC Press, 2012.
- 2 Paul S Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical report, Carnegie-Mellon University of Pittsburgh, 1997.

Topological stability of kinetic k -centers

Ivor Hoog v.d., Marc van Kreveld¹

Dept. of Information and Computing Sciences, Utrecht University, The Netherlands
[i.d.vanderhoog|m.j.vankreveld]@uu.nl

Wouter Meulemans, Kevin Verbeek, Jules Wulms²

Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands
[w.meulemans|k.a.b.verbeek|j.j.h.m.wulms]@tue.nl

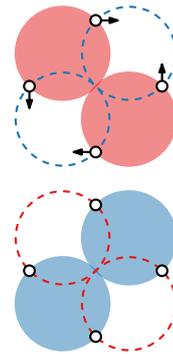
1 Introduction

The k -center problem or facility location problem asks for a set of k disks that cover a given set of n points, such that the maximum radius of all the disks is as small as possible. Since the introduction of the k -center problem by Sylvester [3] in 1857, the problem has been widely studied and has found many applications in practice.

In recent decades there has been an increased interest, especially in the computational geometry community, to study problems for which the input points are moving, including the k -center problem. These problems are typically studied in the framework of *kinetic data structures* [1], where the goal is to efficiently maintain the (optimal) solution to the problem as the points are moving.

In many practical applications, for example if the disks are represented physically, or if the disks are used for visualization, the disks should move smoothly as the points are moving smoothly. As the optimal k -center (for $k \geq 2$) may exhibit discontinuous changes as points move (see figure), we need to resort to approximations to guarantee stability.

Recently, Meulemans et al. [2] introduced a new framework for algorithm stability, which includes the definition of *topological stability*. An algorithm is topologically stable if its output behaves continuously (albeit with arbitrary speed) as the input is changing. The topological stability ratio ρ_{TS} of a problem is then defined as the ratio between the quality of a topologically stable solution and an optimal but unstable solution. In [2] bounds on ρ_{TS} are given for kinetic Euclidean minimum spanning trees, using various ways of enforcing continuity.



In this abstract we prove the following theorem on the topological stability of k -center.

► **Theorem 1.** *For the k -center problem it holds that $2 \sin(\frac{\pi(k-1)}{2k}) \leq \rho_{\text{TS}} \leq 2$ for $k \geq 2$.*

2 Bounds on topological stability

As illustrated above, some point sets have more than one optimal solution. If we can *transform* an optimal solution into another, by growing the covering disks at least/at most a factor r , we immediately obtain a lower/upper bound of r on the topological stability. The transformations or *morphs* allow (the centers of the) disks to move and radii to change continuously, as long as the points are covered at all times. We first introduce some tools to help us model and reason about these transformations.

¹ I.H. supported by the Netherlands Organisation for Scientific Research (NWO) through project no 614.001.504.

² J.W. and W.M. are (partially) supported by the Netherlands eScience Center (NLeSC) under grant number 027.015.G02. K.V. is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.021.541.

2:2 Topological stability of kinetic k -centers

2-colored intersection graphs. Consider a point set P and two sets of k disks, such that each set covers all points in P : we use R to denote the one set (red) and B to denote the other set (blue). We now define the *2-colored intersection graph* $G_{R,B} = (V, E)$: each vertex represents a disk ($V = R \cup B$) and is either red or blue; E contains an edge for each pair of differently colored, intersecting disks. A 2-colored intersection graph always contains equally many red nodes and blue nodes by definition and both colors must cover all points: there may be points only in the area of intersection between a blue and red disk. In the remainder, we use *intersection graph* to refer to 2-colored intersection graphs.

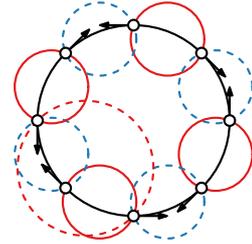
► **Lemma 2.** *Let sets R and B of k disks each cover a point set P . If $G_{R,B}$ is a forest, then R can morph onto B without increasing the disk radius, while covering all points in P .*

Proof. From a counting argument, using $|R| = |B|$, it follows that we can always find a red leaf in $G_{R,B}$. This red leaf can then morph onto its blue neighbor. This effectively removes these two nodes from the intersection graph, since the blue disk is fully covered by the red disk; repeating this argument gives a morph from R onto B . ◀

Proving Theorem 1. We are now ready to deal with the k -center problem. The upcoming lemmata each prove one part.

► **Lemma 3.** *For the k -center problem it holds that $\rho_{\text{TS}} \geq 2 \sin(\frac{\pi(k-1)}{2k})$ for $k \geq 2$.*

Proof. Consider a set of $2k$ points, which are the corners of a regular $2k$ -gon with unit radius, i.e., equidistantly spread along the boundary of a unit circle. There are exactly two optimal solutions R and B on these points, for which $G_{R,B}$ forms a cycle (see figure). To morph from R to B , one of the red disks r_1 has to grow to cover the intersection of an adjacent blue disk b with the other (red) neighbor r_2 of b (see dashed red disk). The diameter of the disks in our optimal solution equals the length of a side of this regular $2k$ -gon, hence r_1 has to grow with a factor $2 \sin(\frac{\pi(k-1)}{2k})$. Once r_1 has grown to overlap the intersection between a blue disk and r_2 , r_2 no longer has to cover the points in the intersection and can be treated as a degree-1 vertex in $G_{R,B}$. Since that makes $G_{R,B}$ a tree, we can apply Lemma 2.



If we can show that a set of *moving* points actually forces this swap to happen, the desired bound on the topological stability follows from the above argument. We can place points moving on tangents of the circle defining the $2k$ points, to arrive at the described situation at a time t , while ensuring that a swap before or after t would be only more costly. ◀

► **Lemma 4.** *For the k -center problem it holds that $\rho_{\text{TS}} \leq 2$ for $k \geq 2$.*

Proof. Consider a moment in time t where there are two optimal solutions; let R denote the optimal solution at $t - \varepsilon$ and B the optimal solution at $t + \varepsilon$ for arbitrarily small $\varepsilon > 0$. Let C be the maximum radius of the disks in R and in B and let $G_{R,B}$ describe their intersections. First we make a maximal matching between red and blue vertices that are adjacent in $G_{R,B}$. The intersection graph of the remaining red and blue disks has no edges, so we match these red and blue disks in any way. All the red disks that are matched to blue disks they already intersect grow to overlap their initial disk and the matched blue disk. Now the remaining red disks can safely move to the blue disks they are matched to, and adjust their radii to fully cover the blue disks. Finally, to finish the morph all red disks shrink. When all red disks are overlapping blue disks, the maximum of their radii is at most $2C$. ◀

References

- 1 Julien Basch, Leonidas J. Guibas, and John Hershberger. Data structures for mobile data. *Journal of Algorithms*, 31(1):1–28, 1999.
- 2 Wouter Meulemans, Bettina Speckmann, Kevin Verbeek, and Jules Wolms. A framework for algorithm stability. In *Proc. 13th Latin American Symposium on Theoretical Informatics*, LNCS 10807, pages 805–819, 2018.
- 3 James Sylvester. A question in the geometry of situation. *Quarterly Journal of Mathematics*, 1:79, 1857.

A KDS for Discrete Morse-Smale Complexes*

T. Ophelders, W. Sonke, B. Speckmann, and K. Verbeek

Dep. of Mathematics and Computer Science, TU Eindhoven, The Netherlands
[t.a.e.ophelders|w.m.sonke|b.speckmann|k.a.b.verbeek]@tue.nl

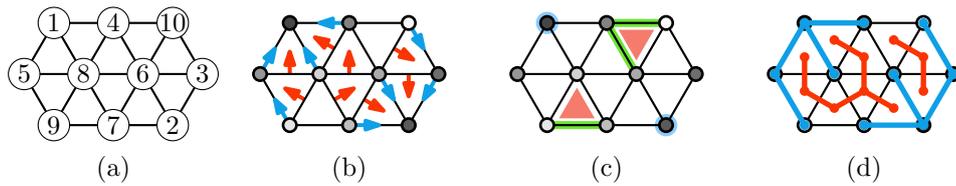
Introduction. Consider a two-dimensional surface S with a height function $h : S \rightarrow \mathbb{R}$. The Morse-Smale complex (MS-complex) of T is a topological complex that provides information about the features of the height function on the terrain. It consists of the critical points (minima, saddles and maxima) of h in T , together with steepest-descent paths from saddles to minima and steepest-ascent paths from saddles to maxima. In the continuous case, the MS-complex is well-defined if h is a Morse function: each critical point of h has a distinct height, and certain types of degeneracies do not occur. To allow computing MS-complexes on real-world measurement data, which typically is discrete, several extensions of Morse functions to the discrete case have been studied. An extensive explanation of the most prominent of those, *discrete Morse theory*, is provided by Forman [2]. Based on discrete Morse theory, there have been several approaches to define discrete MS-complexes. In this work we focus on the discrete MS-complex defined by Shivashankar *et al.* [3]. We present a kinetic data structure (KDS) for this MS-complex. That is, we consider a height function h that changes over time, and provide a data structure to track the MS-complex throughout this movement. This can be used to efficiently analyze time-varying data.

Discrete MS-complex. The discrete MS-complex computed by Shivashankar *et al.* is defined by a discrete gradient field, which is a set of *gradient pairs*. While gradient fields are defined for any cell complex, to simplify the presentation, we assume here that the input is a triangulated (two-dimensional) terrain K . In this setting, there are two types of gradient pairs: those between vertices and edges, and those between edges and faces. Specifically, a vertex v_1 is paired with the edge $\{v_1, v_2\}$ towards its lowest neighbor v_2 . (If no neighbor lower than v_1 exists, then v_1 is not paired with an edge.) Furthermore consider the triangles $\{v_1, v_2, v_3\}$ and $\{v_1, v_2, v'_3\}$ incident to an edge $\{v_1, v_2\}$. This edge is paired with the face $\{v_1, v_2, v_{\min}\}$ where v_{\min} is the lowest vertex among v_3 and v'_3 . (If none of v_3 and v'_3 are lower than both v_1 and v_2 , then $\{v_1, v_2\}$ is not paired to a face.) A vertex, edge or face that is not paired with anything is called *critical*; critical vertices, edges and faces are minima, saddles and maxima, respectively (see Fig. 1a–c). The *ascending manifold* of a minimum v is obtained by traversing reversed gradient pairs, starting from v . The *descending manifold* of a maximum v is obtained by traversing gradient pairs, starting from v .

KDS. We aim to construct a KDS to maintain the minima, saddles and maxima, and the ascending and descending manifolds as the vertices continuously change their height. We assume that at no point in time, three vertices have the same height. Our data structure is inspired by the one proposed by Agarwal *et al.* for maintaining contour trees kinetically [1]. Like Agarwal *et al.* we use *link-cut trees*, a data structure that stores a forest of rooted trees dynamically, supporting edge insertions and deletions. Furthermore, the root of each tree can be set and for any vertex the root of its tree can be found. All of these operations take logarithmic time.

To maintain the ascending and descending manifolds, we use two link-cut trees, T_{\downarrow} and T_{\uparrow} (see Fig. 1d). T_{\downarrow} represents the vertex-to-edge gradient pairs. Specifically, T_{\downarrow} contains a vertex for each vertex in K , and it contains the edge $\{v_1, v_2\}$ for each vertex-to-edge gradient pair $(v_1, \{v_1, v_2\})$. In the static setting discussed by Shivashankar *et al.*, the ascending

* The authors are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.023.208 (T.O., W.S., and B.S.) and no. 639.021.541 (K.V.).



■ **Figure 1** (a) A terrain with vertex heights; (b) vertex-edge (blue) and edge-face (red) gradient pairs; (c) minima (blue), saddles (green) and maxima (red); (d) T_{\downarrow} (blue) and T_{\uparrow} (red).

manifolds are computed by a BFS starting from each minimum, traversing reversed gradient pairs. Such a BFS corresponds to traversing one complete tree in T_{\downarrow} . Hence, each tree in T_{\downarrow} represents an ascending manifold; we ensure that its minimum is the root of the tree.

T_{\uparrow} represents the edge-to-face gradient pairs. Specifically, T_{\uparrow} contains a vertex for each face in K , and it contains the edge $(\{v_1, v_2, v_3\}, \{v_1, v_2, v_3'\})$ for each edge-to-face gradient pair $(\{v_1, v_2\}, \{v_1, v_2, v\})$. Again, as this mirrors the BFS in the static setting, each tree in T_{\uparrow} represents a descending manifold; we ensure that its maximum is the root of the tree.

Event handling. We first show how to maintain the set of vertex-edge gradient pairs; that is, T_{\downarrow} . Changes in the vertex-edge gradient pairs happen because the lowest neighbor of a vertex changes. Specifically, when the lowest neighbor of vertex v changes, v needs to be paired with its incident edge that is now lowest. To track this information, we store a tournament tree for each vertex v , to maintain its lowest neighbor. This tournament tree contains v 's neighboring vertices and v itself. This leads to three types of events: the lowest neighbor can move from one neighbor v_1 to another neighbor v_2 (in which case we update T_{\downarrow} by deleting $\{v, v_1\}$ and inserting $\{v, v_2\}$), the lowest neighbor can move from a neighbor v_1 to v itself (in which case we delete $\{v, v_1\}$ from T_{\downarrow}), or the lowest neighbor can move from v to a neighbor v_1 (in which case we insert $\{v, v_1\}$ into T_{\downarrow}). Several such events can happen at the same time, in which case we handle them one by one. To avoid adding cycles to T_{\downarrow} , we first execute all edge deletions, and then all insertions. Similarly we maintain T_{\uparrow} , by maintaining for each edge $\{v_1, v_2\}$ which of v_1, v_2, v_3 and v_3' is the lowest. After an event has been handled, we can locally determine which vertices, edges and faces in the neighborhood are minima, saddles and maxima, respectively, and mark them as such.

Running time. Because an event influences only the neighborhood of a single vertex or face, per event only a constant number of link / cut operations need to be carried out. Assuming the maximum vertex degree in K is bounded by a constant, events can be processed in $O(\log n)$ time each. Hence, if there are k changes to the MS-complex, our KDS can compute those in $O(k \log n)$ time in total.

References

- 1 P. K. Agarwal, T. Mølhave, M. Revsbæk, I. Safa, Y. Wang, and J. Yang. Maintaining contour trees of dynamic terrains. In *Proc. 31st International Symposium on Computational Geometry*, pages 796–811, 2015.
- 2 R. Forman. A user's guide to discrete Morse theory. *Séminaire Lotharingien de Combinatoire*, 48:1–35, 2002.
- 3 N. Shivashankar, Senthilnathan M, and V. Natarajan. Parallel computation of 2D Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1757–1770, 2012.

Spanners for Topological Summaries

Michael Kerber

Graz University of Technology
[Kopernikusgasse 24, 8010, Graz, Austria]
kerber@tugraz.at
 <https://orcid.org/0000-0002-8030-9299>

Arnur Nigmatov

Graz University of Technology
[Kopernikusgasse 24, 8010, Graz, Austria]
nigmatov@tugraz.at

Abstract

Given n persistence diagrams, can we obtain a close approximation of the induced metric space without computing all pairwise distances? We address this question by a spanner construction in the space of persistence diagrams. For that, we adopt the practically efficient cover tree construction to the case of approximate distance computation and construct a well-separated pair decomposition out of the cover tree. Because the space of persistence diagrams is of high doubling dimension, our approach does not yield worst-case guarantees, even under quite favorable assumptions on the input. However, we show that in practice, the number of distance computations drops significantly for clustered data. Our results and methodology also carry over to the case of Reeb graphs.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures → Computational geometry

Keywords and phrases spanners, doubling dimension, persistence diagram, Wasserstein distance

Funding This work is supported by the Austrian Science Fund (FWF) grant number P 29984-N35

1 Summary

In topological data analysis, it is often necessary to compute all pairwise distances between topological summaries like persistence diagrams or Reeb graphs. Unlike in the standard example of Euclidean space, a single distance computations cannot be assumed to be cheap; for instance the complexity of computing the bottleneck distance between persistence diagram with N points is $O(N^{1.5} \log N)$ with the best known algorithms, and even higher for the equally-popular Wasserstein distance. Moreover, in the case of Wasserstein distances, efficient implementations only exist for computing relative approximations of the distance [3]. The situation is even worse for Reeb graphs, where for none of the proposed distances (e.g. [2]), a polynomial-time (approximation) algorithm is known. On the other hand, such topological summaries are frequently used in practice to classify and cluster data. Hence, it is worthwhile to reduce the number of distance computations as much as possible.

In this note, we focus on the case of persistence diagrams with Wasserstein distance for brevity. Our approach follows the standard strategy for finite metric spaces. We use a cover tree [1] to obtain a hierarchical decomposition of the metric space (net trees yield slightly better theoretical guarantees, but we decided for cover trees because efficient implementations exist). A technical obstacle is posed by the fact that we can only compute distances up to an approximation factor – however, we can adapt the definition of cover trees to handle that case as well. Out of the cover tree, we construct a well-separated pair decomposition

(WSPD) of the metric space. While the algorithm is identical to the Euclidean case (using compressed quad-trees), there seems no preceding case where WSPDs have been constructed from cover trees in the literature. We show that an ϵ -WSPD of size $O(c^{14}n(1/\epsilon)^{\log c})$ can be constructed from a cover tree where c is the expansion constant of the metric space [1]. An ϵ -WSPD defines an 2ϵ -spanner of the metric space using standard methods.

The success of our strategy (in theory and practice) depends on the dimensionality of our metric space, expressed by the expansion constant c above. Note that $\log c$ is lower bounded by the doubling dimension Δ of the metric space. Unfortunately, the space of persistence diagrams has infinite doubling dimension, and when restricting to n samples, Δ can be as high as $\log n$. Even worse, we give an example of n input functions that can be isometrically embedded on a line (and hence have doubling dimension 1), such that the corresponding persistence diagrams still have dimension $\log n$. Hence, we cannot expect improved worst-case results over the naive approach to compute all $\binom{n}{2}$ distances.

On the other hand, the above analysis does not imply that the spanner construction is useless in every practical instance. Due to the relatively high cost of the distance computations, the overhead of computing a cover tree and a spanner out of it is relatively low. Hence, even if the construction ends up computing, say, 99% of all pairwise distances, the performance penalty is negligible. We show that in some situations, however, the gain is substantial. In the first row of Table 1, we generated diagrams that are densely clustered around centers which are relatively far apart from each other. This is clearly a “cherry-picked” situation for WSPD construction, and indeed, we see that a 0.04-spanner can be obtained computing only 6 percent of the pairwise distances.

We also generate real persistence diagrams of the shapes from the McGill shape benchmark (<http://www.cim.mcgill.ca/~shape/benchMark/>) to which we added some random noise. Here, depending on the parameters, we can either get a 2-approximation while still computing less than 6 percent of the pairwise distances, or, if we want more accurate results, we can get 23 % error and still reduce the number of distance computations by a factor of 2.

Dataset	# diagrams	Cover tree fraction	WSPD fraction	Real relative error
Synthetic	781	0.021514	0.055406	0.0371734
Real	732	0.014577	0.051681	0.822682
Real	732	0.016027	0.515224	0.225174

■ **Table 1** Results of our experiments. Cover tree fraction shows the portion of distances that were really computed when building the cover tree; WSPD fraction shows the portion of distance that were computed when building WSPD; real relative error is the largest error on all pairwise distances.

References

- 1 A. Beygelzimer, S. Kakade, and J. Langford. Cover Trees for Nearest Neighbor. In *ICML*, pages 97–104. ACM, 2006.
- 2 Vin De Silva, Elizabeth Munch, and Amit Patel. Categorized Reeb Graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016.
- 3 M. Kerber, D. Morozov, and A. Nigmetov. Geometry Helps to Compare Persistence Diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4, 2017.

A Kernel for Multi-parameter Persistence

René Corbet

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, 8010 Graz, Austria
corbet@tugraz.at
 <https://orcid.org/0000-0003-3148-4039>

Ulderico Fugacci

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, 8010 Graz, Austria
fugacci@tugraz.at
 <https://orcid.org/0000-0003-3062-997X>

Michael Kerber

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, 8010 Graz, Austria
kerber@tugraz.at
 <https://orcid.org/0000-0002-8030-9299>

Claudia Landi

Department of Engineering Sciences and Methods, University of Modena and Reggio Emilia
Via Amendola 2, Pad. Morselli, 42100 Reggio Emilia, Italy
claudia.landi@unimore.it
 <https://orcid.org/0000-0001-8725-4844>

Bei Wang

Scientific Computing and Imaging Institute, University of Utah
72 South Central Campus Drive, Salt Lake City, Utah 84112, USA
beiwang@sci.utah.edu
 <https://orcid.org/0000-0002-9240-0700>

Abstract

Topological data analysis and its main invariant, *persistent homology*, provide a toolkit for computing topological information of noisy spaces. *Kernels* for one-parameter persistent homology have been established to connect persistent homology with machine learning techniques. We contribute a kernel construction for multi-parameter persistence by integrating a one-parameter kernel along straight lines and prove stability for a wide range of useful one-parameter kernels.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures, Computing methodologies → Kernel methods

Keywords and phrases Persistent homology, Kernel methods, Multi-parameter Persistence

Funding The first three authors are supported by the Austrian Science Fund (FWF) grant number P 29984-N35. The fifth author is supported by NSF-IIS-1513616 and NIH-R01-1R01EB022876-01.

Acknowledgements This work was initiated during the Dagstuhl workshop “Topology, Computation and Data Analysis” (17292).

Summary

It is well-known that the set of persistence diagrams forms a metric space, for instance using the bottleneck distance. However, based on the complicated nature of this space, any form of statistical analysis (e.g., computing averages) is a difficult task. A trend in the last years has been to embed persistence diagrams into a larger space where such statistical methods can be applied directly. Among such techniques, embeddings into a Hilbert space have been proposed, allowing the definition of an inner product (also called a *kernel*) of persistence diagrams [2]. The rough idea is to replace each off-diagonal point in the persistence diagram by a Gaussian peak and obtain a *feature map* $\mathbb{R}^2 \rightarrow \mathbb{R}$ as the sum of the Gaussians. Then, the kernel is simply defined as the inner product in the L^2 -space over \mathbb{R}^2 .

On the other hand, the theory of multi-parameter persistence poses mathematical challenges, mainly due to the fact that a “persistence diagram” does not exist as in the one-parameter counterpart. Nevertheless, several notions of distance have been proposed for multi-parameter persistence modules. Among those methods is the *matching distance*: the basic idea is to compare two multifiltrations along linear *slices* through the persistence space. Along every such slice, one obtains two one-parameter modules which can be compared using the bottleneck distance. The supremum of all these (appropriately weighted) distances is the matching distance. A strong point of this method is the possibility to approximate the distance in polynomial time by using a subsample of slices [1].

We contribute the first kernel for multi-parameter persistence by a combination of the two described techniques. We restrict to the case of two parameters for brevity in this note. On a high level, we construct a feature map $\mathbb{R}^4 \rightarrow \mathbb{R}$ from a persistence module, and define the kernel via the L^2 -space over \mathbb{R}^4 . For the definition of the feature map, we consider slices of the bifiltered module; on each slice, we construct a feature map in $L^2(\mathbb{R}^2)$ (e.g., using the feature map from [2]), and we combine all these feature maps into one map, where the collection of lines yields two additional degrees of freedom.

To ensure that the constructed feature map is L^2 -integrable, we require a global upper bound $N \in \mathbb{N}$ such that the persistence diagram along every slice has at most N off-diagonal points. Moreover, we restrict our attention to a bounded rectangle R in persistence space – while we can generalize this assumption, we claim that a canonical area of interest can be identified in most realistic scenarios. Furthermore, instead of the feature map from [2], our construction works as well with various other embeddings proposed in the literature.

Our constructed feature map gives rise to a stable distance. More precisely, let $F, G : X \rightarrow \mathbb{R}^2$ be functions from a topological space X . The sublevel sets of F and G define two-parameter persistence modules, and our construction yields feature maps Φ_F, Φ_G which are L^2 -integrable under mild assumptions on F and G . With N and R as above, we have

$$\|\Phi_F - \Phi_G\|_{L^2} \leq C \cdot N \cdot \text{area}(R) \cdot \|F - G\|_\infty,$$

where C is an absolute constant and $\|F - G\|_\infty = \sup_{x \in X} \|F(x) - G(x)\|$.

References

- 1 Silvia Biasotti, Andrea Cerri, Patrizio Frosini, and Daniela Giorgi. A new algorithm for computing the 2-dimensional matching distance between size functions. *Pattern Recognition Letters*, 32(14):1735 – 1746, 2011.
- 2 Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4741–4748, 2015.

The Dynamic Wrap Complex

Katharina Ölsböck¹

IST Austria (Institute of Science and Technology Austria), Am Campus 1,
3400 Klosterneuburg, Austria
katharina.oelsboeck@ist.ac.at

1 Introduction

The Wrap complex [1, 4] is a useful tool for shape reconstruction. It is a filtered subcomplex of the Delaunay triangulation. In many applications, the set of points on which the Delaunay triangulation is computed changes dynamically: points are inserted, deleted, and moved. Dynamic algorithms have been proposed and implemented to efficiently update the Delaunay triangulation [3], however, no work has been done on updating the Wrap complex locally. We propose an algorithm that updates the Wrap complex by only recomputing the filtration values of simplices affected by the operation.

In this work, we first give a definition of the Wrap complex, followed by a discussion of the algorithm to update it locally. It has been implemented in C++ for points in \mathbb{R}^2 and \mathbb{R}^3 , using the CGAL-library [2] for parts of the computation.

2 Background

Given a finite set of points, $X \subseteq \mathbb{R}^d$, in general position, the *Delaunay triangulation*, $\text{Del}(X)$, consists of all simplices $\sigma \subseteq X$, for which there exists a sphere with the points of σ on its boundary and no points of X inside (an *empty circumsphere* of σ). The *Delaunay radius function*, ρ , assigns to each simplex of $\text{Del}(X)$ the radius of its smallest empty circumsphere. In [1], it is shown that ρ is a generalized discrete Morse function, which implies that there is a partition of $\text{Del}(X)$ into intervals $[\sigma, \tau] := \{\nu \mid \sigma \subseteq \nu \subseteq \tau\}$ so that for simplices $\sigma \subseteq \tau$ it holds that $\rho(\sigma) \leq \rho(\tau)$, with equality iff they are in the same interval. An interval is called *singular* if it only contains one simplex, which we then call *critical*.

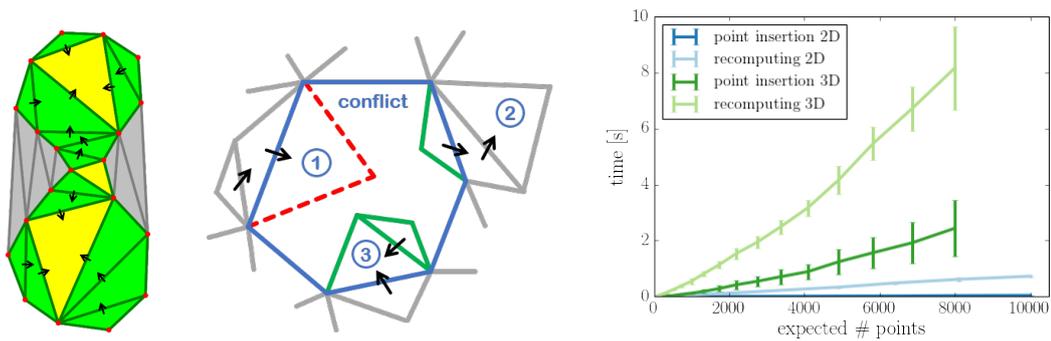
We define a directed graph whose nodes are the intervals. There is an arc from interval a to interval b iff there are simplices $\sigma \in a, \tau \in b$ with $\sigma \subseteq \tau$. The *lower set* of a node, μ , are the nodes from which μ can be reached along a directed path in the graph. We call μ a *descendant* of the nodes in its lower set. The *Wrap complex*, $\text{Wrap}_r(X)$, is the union of the lower sets of all critical simplices σ with $\rho(\sigma) \leq r$; see Figure 1. The *Wrap radius* of a simplex $\sigma \in \text{Del}(X)$ is the smallest r so that $\sigma \in \text{Wrap}_r(X)$.

3 Algorithm

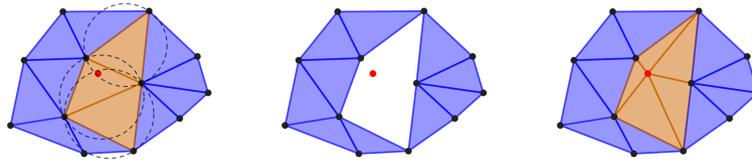
The insertion of a new point into the Delaunay triangulation affects a simplex only if the point is in its smallest empty circumsphere. We call these simplices the *conflict zone* of the point; see Figure 2. After the point insertion, all new simplices are incident to the new point. For the deletion of a point, it is exactly the other way round.

After a dynamic update of the Delaunay triangulation, only the Delaunay radii in the interior and on the boundary of the conflict zone may change. The Wrap radii, however, can change in a bigger region. The changes in the conflict zone might affect simplices outside,

¹ This research is partially supported by the DFG Collaborative Research Center TRR 109, ‘Discretization in Geometry and Dynamics’, through grant no. I02979-N35 of the Austrian Science Fund (FWF).



■ **Figure 1** Illustration of the Wrap complex (*left*) and of the local update steps (*middle*). Running times for a Poisson point process, averaged over 100 runs (*right*).



■ **Figure 2** Conflict zone for insertion (*left to right*) and deletion (*right to left*) of a point.

namely those in the lower sets of simplices on the boundary of the conflict zone. In order to allow efficient updates, every non-singular interval stores references to all its critical descendants. The critical descendant of smallest radius determines the Wrap radius of the interval. After recomputing the intervals inside and on the boundary of the conflict zone and updating the corresponding arcs in the graph, we perform the following steps to update the sets of critical descendants and thus also the Wrap radii; see Figure 1: First, we remove the references to deleted critical descendants from the boundary intervals and their lower sets. Second, references to critical descendants outside of the conflict zone are added to the part of their lower sets that has changed. Third, we add the new singular intervals as critical descendants to their lower sets. We always traverse the lower sets recursively, following the reversed arcs, and stop when we reach singular intervals, since they and their lower sets do not depend on the more distant critical descendant.

In addition to the update times of the Delaunay triangulation, the running time for a dynamic update of the Wrap complex only depends on the size of the conflict zone and the lower sets of the boundary. If this region is small compared to the entire complex, we achieve much faster running times than for recomputing everything; see Figure 1.

References

- 1 U. BAUER AND H. EDELSBRUNNER. The Morse theory of Čech and Delaunay complexes. *Trans. Amer. Math. Soc.* **369** (2017), 3741–3762.
- 2 CGAL. Computational Geometry Algorithms Library. <http://www.cgal.org>.
- 3 O. DEVILLERS, S. MEISER AND M. TEILLAUD. Fully dynamic Delaunay triangulation in logarithmic expected time per operation. *Computational Geometry* **2**(2) (2002), 55–80.
- 4 H. EDELSBRUNNER. Surface reconstruction by wrapping finite point sets in space. In *Discrete and Computational Geometry. The Goodman–Pollack Festschrift*, 379–404, eds. B. Aronov, S. Basu, J. Pach and M. Sharir, Springer-Verlag, 2003.

Classification of filtered chain complexes

B. Giunti¹

University of Pavia

barbara.giunti01@universitadipavia.it

 <https://orcid.org/0000-0002-3500-8286>

W. Chacholski

KTH of Stockholm

wojtek@kth.se

 <https://orcid.org/0000-0001-5459-3161>

C. Landi

University of Modena and Reggio Emilia

clandi@unimore.it

 <https://orcid.org/0000-0001-8725-4844>

Abstract

Persistent homology of the Vietoris-Rips complex has proven to be an efficient tool to extract topological information from finite metric spaces. However, homology is a drastic simplification and in certain situations might remove too much information. This prompts us to consider filtered chain complexes. With this goal, we define and list all and only possible indecomposables of filtered chain complexes, and we provide a structure theorem for enumerating them. We present an algorithm to compute the decomposition whose implementation is underway.

2012 ACM Subject Classification Mathematics of computing → Continuous mathematics → Topology → Algebraic topology; Theory of computation → Randomness, geometry and discrete structures → Computational geometry

Keywords and phrases Topological Data Analysis, interval spheres, decomposition algorithm

1 Decomposition of filtered chain complexes

Our goal is to classify compact *filtered chain complexes*, which are functors $F : \mathbf{N} \rightarrow \mathbf{Ch}$ assigning an inclusion to any morphism (\mathbf{N} is the poset of natural number, and \mathbf{Ch} is the category of chain complexes of k -vector spaces). A filtered chain complex is compact if it is finitely generated and it eventually maps all morphisms to isomorphisms. For example, the Vietoris-Rips complex of a finite metric space yields a compact filtered chain complex. Our structure theorem says all such complexes are sum of the following indecomposables.

► **Definition 1.** Let $0 \leq b < \infty$, $b \leq d \leq \infty$. A $[b, d]$ -*n*-sphere (interval sphere of dimension n) is a filtered chain complex isomorphic to:

$$\begin{array}{cccccccccccc}
 i=0 & & \dots & & i=b-1 & & i=b & & \dots & & i=d-1 & & i=d & & \dots \\
 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \dots \\
 \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow & & \\
 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & k & \xrightarrow{\mathbf{1}} & \dots & h=n+1 \\
 \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow & & \\
 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & k & \xrightarrow{\mathbf{1}} & \dots & \xrightarrow{\mathbf{1}} & k & \xrightarrow{\mathbf{1}} & k & \xrightarrow{\mathbf{1}} & \dots & h=n \\
 \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow & & \\
 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \dots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \dots
 \end{array}$$

¹ Corresponding author, financially supported by INFN and INdAM

Homology transforms filtered chain complexes into persistence modules. If $b < d$, the n th-homology of the $[b, d]$ - n -sphere is the interval persistence module $\mathbb{I}[b, d]$. If $b = d$, the corresponding sphere has trivial homology. Homology forgets information that we believe not necessarily should be regarded as noise. Chain complexes retain not only homological non-trivial information, but also geometrical content encoded by contractible parts. We hope this information can be used for more accurate analysis of data.

1.1 Structure theorem

\mathbf{FCh} denotes the category of compact filtered chain complexes, and for $0 \leq b < \infty$, $b \leq d \leq \infty$ $\mathcal{S}_{b,d}$ is the set of all $[b, d]$ - n -spheres for any $n \geq 0$. Define a hierarchy of full subcategories:

$$\begin{array}{ccccccc}
 \mathbf{FCh} & \xleftarrow{\mathcal{S}_{0,0}^\perp} & \mathbf{FCh}_{0,0} & \xleftarrow{\mathcal{S}_{0,1}^\perp} & \mathbf{FCh}_{0,1} & \xleftarrow{\mathcal{S}_{0,2}^\perp} & \dots \xleftarrow{\dots} \mathbf{FCh}_{0,\infty} \xleftarrow{\dots} \\
 & & & & & & \\
 & & \mathbf{FCh}_{1,0} & \xleftarrow{\mathcal{S}_{1,1}^\perp} & \mathbf{FCh}_{1,1} & \xleftarrow{\mathcal{S}_{1,2}^\perp} & \dots \xleftarrow{\dots} \mathbf{FCh}_{1,\infty} \xleftarrow{\dots} \\
 & & & & & & \\
 & & & & \mathbf{FCh}_{2,1} & \xleftarrow{\mathcal{S}_{2,2}^\perp} & \dots \xleftarrow{\dots} \mathbf{FCh}_{2,\infty} \dots
 \end{array}$$

where $\mathbf{F}' \xleftarrow{\mathcal{S}_{b,d}^\perp} \mathbf{F}$ denotes taking the orthogonal complement \mathbf{F}' of $\mathcal{S}_{b,d}$ in \mathbf{F} , and $\mathbf{FCh}_{b,\infty} := \bigcap_{b \leq i < \infty} \mathbf{FCh}_{b,i}$, for $0 \leq b < \infty$. Such an orthogonal complement is by definition the full subcategory in \mathbf{F} whose objects do not receive any monomorphism from objects in $\mathcal{S}_{b,d}$.

► **Theorem 2.** *Non-trivial filtered chain complexes of \mathbf{FCh} decompose uniquely, up to permutations and isomorphisms, into finite direct sums of interval spheres.*

► **Corollary 3.** *Each compact filtered chain complex is completely described by a finite multiset of points in the extended plane.*

► **Corollary 4.** *Any compact filtered chain complex of \mathbf{FCh} is indecomposable if and only if it is isomorphic to a $[b, d]$ - n -sphere for some $0 \leq n$, $0 \leq b < \infty$, $b \leq d \leq \infty$.*

1.2 Decomposition algorithm

We can devise an algorithm that splits out the interval spheres from a compact filtered chain complex $F = (C_\bullet^0, \partial_\bullet^0) \subseteq (C_\bullet^1, \partial_\bullet^1) \subseteq \dots$, according to the hierarchy of Thm 3.

Input: Boundary matrices ∂_h^i , for $0 < h$, $0 \leq i$.

Output: Number and type of indecomposables.

- for each degree h :
 - for each filtration step i :
 - for each non-zero column c of ∂_h^i , let b be the smallest index in the filtration for which c is in the image of F ($b \leq i$):
 - ◊ reduce all the columns after column c
 - ◊ delete column c and the row of its first non-zero element
 - ◊ delete the corresponding row in the boundary matrix ∂_{h+1}^i
 - increase the number of $[b, i]$ - $(h - 1)$ -spheres by one
 - the boundaries of the obtained filtered chain complex $\bar{C}_\bullet^0 \subseteq \bar{C}_\bullet^1 \subseteq \dots$ are zero
 - the number of $[i, \infty)$ - \bullet -spheres is given by $\dim(\bar{C}_\bullet^i) - \dim(\bar{C}_\bullet^{i-1})$.

The crucial part in this algorithm are the \diamond steps: they achieve the decomposition of F as $F \oplus [b, i]$ -sphere. We note that the algorithm intrinsically skips computation on non-essential rows and reduces the boundary matrices in increasing dimension.

Strong Collapse for Persistence *

Jean-Daniel Boissonnat

DataShape, INRIA, Sophia-Antipolis, France

Siddharth Pritam

DataShape, INRIA, Sophia-Antipolis, France

siddharth.pritam@inria.fr

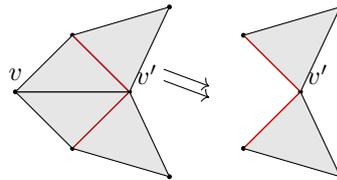
Divyansh Pareek

DataShape, INRIA, Sophia-Antipolis, France

1 Introduction

In this abstract, we introduce a new approach to simplify the complexes of an input sequence which uses the notion of strong collapse introduced by J. Barmak and E. Miniam [1]. Specifically, our approach can be summarized as follows: Given a sequence $\mathcal{Z} : \{K_1 \xrightarrow{f_1} K_2 \xleftarrow{g_2} K_3 \xrightarrow{f_3} \dots \xrightarrow{f_{(n-1)}} K_n\}$ of simplicial complexes K_i connected through simplicial maps $\{\xrightarrow{f_i}$ or $\xleftarrow{g_j}\}$. We independently strong collapse the complexes of the sequence to reach a sequence $\mathcal{Z}^c : \{K_1^c \xrightarrow{f_1^c} K_2^c \xleftarrow{g_2^c} K_3^c \xrightarrow{f_3^c} \dots \xrightarrow{f_{(n-1)}^c} K_n^c\}$, with *induced simplicial maps* $\{\xrightarrow{f_i^c}$ or $\xleftarrow{g_j^c}\}$ (defined in Section 4). The complex K_i^c is called the **core** of the complex K_i and we call the sequence \mathcal{Z}^c the **core sequence** of \mathcal{Z} . We show that one can compute the persistent homology (PH) of the sequence \mathcal{Z} by computing the PH of the core sequence \mathcal{Z}^c , which is of much smaller size.

We call a vertex v of a simplicial complex K a dominated vertex, if its *link* in K is a join of a vertex v' of K and a subcomplex L of K , i.e a simplicial cone. We remove such v from K by removing all the simplices of K that contained v . Removal of such a vertex v is called an **elementary strong collapse** (see Fig 1) and a sequence of elementary strong collapses is called a **strong collapse**. Strong collapse preserves the homotopy type of the complex. We use this notion of strong collapse to reduce the size of simplicial sequences and to speed up the persistence computation.



■ **Figure 1** Illustration of an *elementary strong collapse*. In the complex on the left, v is dominated by v' . The link of v is highlighted in **red**. Removing v leads to the complex on the right.

Our method has some similarity with the work of Wilkerson et. al. [2] who also use strong collapses to reduce PH computation but it differs in three essential aspects: it is not limited to filtrations (i.e. sequences of nested simplicial subcomplexes) but works for other

* This research has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP/2007- 2013) / ERC Grant Agreement No. 339025 GUDHI (Algorithmic Foundations of Geometry Understanding in Higher Dimensions).

8:2 Strong Collapse for Persistence

\mathcal{X}	1-sphere	2-Annulus	dragon	netw-sc	senate	eleg
PDF	0.65	174.18	69.92	243.86	24.92	10.87
PDT	0.060	0.178	0.553	0.097	0.068	0.165

■ **Table 1** The rows are, from top to down: dataset \mathcal{X} , PD computation time for the original filtration (PDF), Total PD computation time of the tower (PDT). All times are noted in seconds. For the first three datasets, we sampled points randomly from the initial datasets and averaged the results over five trials.

types of sequences like towers and zigzags. It also differs in the way the strong collapses are computed and in the way PH is computed.

A first central observation is that to strong collapse a simplicial complex K , we only need to store its maximal simplices (i.e. those simplices that have no coface). The number of maximal simplices is smaller than the total number of simplices by a factor that is exponential in the dimension of the complex. It is linear in the number of vertices for a variety of complexes. Working only with maximal simplices dramatically reduces the time and space complexities compared to the algorithm of [2]. We prove that the complexity of our algorithm is $\mathcal{O}(v^2 d \Gamma_0 + m^2 \Gamma_0 d)$. Here d is the dimension of the complex, v is the number of vertices, m is the number of maximal simplices and Γ_0 is an upper bound on the number of maximal simplices incident to a vertex. Γ_0 is a small fraction of the number of maximal simplices.

All PH algorithms take as input a full representation of the complexes and their complexity is polynomial in the total number of simplices of the complexes. We thus have to convert the representation by maximal simplices used for the strong collapses into a full representation of the complexes, which takes exponential time in the dimension (of the collapsed complexes). This exponential burden is to be expected since it is known that computing PH is NP-hard when the complexes are represented by their maximal faces. Nevertheless, we demonstrate in this paper that strong collapses combined with known persistence algorithms lead to major improvements over previous methods to compute the PH of a sequence. This is due in part to the fact that strong collapses reduce the size of the complexes on which persistence is computed. It is also due to other facts

- The collapses of the complexes in the sequence can be done independently and in parallel. This is due to the fact that strong collapses can be associated to simplicial maps unlike simple collapses.

- The size of the complexes in a sequence do not grow by much in terms of maximal simplices, as observed in many practical cases. As a consequence, the time to collapse the i -th simplicial complex K_i in the sequence is almost independent of i . For filtrations, this is a neat advantage over methods that use a full representation of the complexes and suffer an increasing cost as i increases.

As a result, our approach is extremely fast and memory efficient in practice as demonstrated by numerous experiments on publicly available data sets. For experiments described in Table 1, we used filtrations of the Rips complex with varying scale parameter to run our experiments. We will describe them in details in the full version of the abstract.

References

- 1 J.A. Barmak, E.G. Minian. Strong homotopy types, nerves and collapses. *Discrete and Computational Geometry* 47(2012), pp. 301-328.

- 2 A. C. Wilkerson, H. Chintakunta and H. Krim. Computing persistent features in big data: A distributed dimension reduction approach. ICASSP - Proceedings (2014) pp. 11-15..



Morse Complexes for Zigzag Persistent Homology

Clément Maria

Inria Sophia Antipolis – Méditerranée
clement.maria@inria.fr

Hannah Schreiber

University of Technology of Graz
hschreiber@tugraz.at

Motivation Algorithms for persistent homology have been successfully optimized, leading to major speed-ups in practice and to the use of topological methods in applied sciences. Zigzag persistence is a generalization of persistent homology with promising theoretical properties. However, persistence optimizations do not seem to adapt to existing implementations [1, 3] of zigzag persistence, hence resulting in a substantial gap in practical performance between the two methods. Indeed, computing zigzag persistence requires one to store a full set of m (co)chains when maintaining the (co)homology of a complex of size m , whereas standard persistence only needs a much smaller subset of (co)chains, allowing optimization techniques such as matrix compression or reduction shortcuts. It is consequently natural to consider optimizations at the level of the complex, as opposed to optimizations at the level of the (co)homology matrix. Discrete Morse theory [2] is a combinatorial method that reduces a complex to a subset of *critical cells* defining its (co)homology, and which has been studied for computing persistent homology [4]. The objective of this work is to use discrete Morse theory in order to improve zigzag persistence computation as well.

Zigzag Filtration and Persistent Homology A *filtration* is a sequence of nested complexes $\mathbb{K}_0 \rightarrow \mathbb{K}_1 \rightarrow \dots \rightarrow \mathbb{K}_m$, where arrows represent inclusion maps.

In a *zigzag filtration*, we also allow deletion of simplices in the sequence, i.e. inclusion map arrows going backward. Therefore we represent a zigzag filtration as follows: $\mathbb{K}_0 \leftrightarrow \mathbb{K}_1 \leftrightarrow \dots \leftrightarrow \mathbb{K}_m$, where \leftrightarrow is either an arrow going forward or going backward. If we apply the homology functor $H(\cdot, \mathbb{F})$, for a fixed field \mathbb{F} , to a zigzag filtration, we obtain a *persistence module*, which can uniquely be decomposed into the direct sum $\bigoplus_{i=1}^s I_{b_i, d_i}$, with the intervals $I_{b,d}$, $b \leq d$, of the form: $\underbrace{0 \xleftrightarrow{0} \dots \xleftrightarrow{0} 0}_{b-1 \text{ times}} \xleftrightarrow{0} \underbrace{\mathbb{F} \xleftrightarrow{\text{id}} \dots \xleftrightarrow{\text{id}} \mathbb{F}}_{d-b+1 \text{ times}} \xleftrightarrow{0} \underbrace{0 \xleftrightarrow{0} \dots \xleftrightarrow{0} 0}_{m-d \text{ times}}$. The aim of the persistence algorithm is to compute this set of pairs (b_i, d_i) , called the *persistence barcode*.

Morse Complex Given a standard filtration $\mathcal{F} = \mathbb{K}_0 \rightarrow \mathbb{K}_1 \rightarrow \dots \rightarrow \mathbb{K}_m$. The algorithm of Mischaikow and Nanda [4] partitions each \mathbb{K}_i into three subsets of simplices \mathcal{A}_i , \mathcal{Q}_i and \mathcal{K}_i , such that the *filtered Morse complex* $\mathcal{A}_0 \rightarrow \mathcal{A}_1 \rightarrow \dots \rightarrow \mathcal{A}_m$, defined on the \mathcal{A}_i with the appropriate boundary map $\tilde{\partial}$, is a filtration with the same barcode as \mathcal{F} . The proof relies on standard algebraic topology methods which nicely adapt to the filtration case, such as chain equivalences. The simplices in \mathcal{Q}_i and \mathcal{K}_i form pairs $(\tau, \sigma) \in \mathcal{Q}_i \times \mathcal{K}_i$, where τ is facet of σ , of "disposable" simplices, not essential in the generation of the homology groups. In practice, efficient heuristics allow us to compute a large number of those pairs, resulting in a rather small Morse complex.

Morse Complexes in Zigzag Persistence Removal of simplices is the major difference in zigzag filtrations. Compared to the non zigzag case, we can have the following operation: $\mathbb{K}_i \xleftarrow{\Sigma} \mathbb{K}_{i+1}$, where Σ is the set of removed simplices. This poses two major questions: how

do we update the partition $\mathcal{A} \sqcup \mathcal{Q} \sqcup \mathcal{K}$ defining the Morse complex, and how do we update the homology basis such that it remains coherent with the whole filtration?

First, we update the Morse complex under the removal of a maximal face $\sigma \in \Sigma$. We know that σ has no coface, thus $\sigma \in \mathcal{A}_i \sqcup \mathcal{K}_i$. If $\sigma \in \mathcal{A}_i$ is critical, the partition $(\mathcal{A}_i \setminus \{\sigma\}) \sqcup \mathcal{Q}_i \sqcup \mathcal{K}_i$ is a valid Morse partition for the complex $\mathbb{K}_i \setminus \{\sigma\}$. If $\sigma \in \mathcal{K}_i$, then it is paired with a simplex $\tau \in \mathcal{Q}_i$, and we distinguish two cases: either $\tau \in \Sigma$, or not. In the first case, both can be simply removed respectively from \mathcal{K}_i and \mathcal{Q}_i , leading to a Morse partition $\mathcal{A}_i \sqcup (\mathcal{Q}_i \setminus \{\tau\}) \sqcup (\mathcal{K}_i \setminus \{\sigma\})$ for the complex $\mathbb{K}_i \setminus \{\tau, \sigma\}$. In the second case, τ can represent homological information and thus has to be transferred from \mathcal{Q}_i to \mathcal{A}_i , leading to the Morse partition $(\mathcal{A}_i \cup \{\tau\}) \sqcup (\mathcal{Q}_i \setminus \{\tau\}) \sqcup (\mathcal{K}_i \setminus \{\sigma\})$ of $\mathbb{K}_i \setminus \{\sigma\}$. It is important to notice that removing faces in $\mathcal{Q} \sqcup \mathcal{K}$ affects the definition of ∂ . Also note that, perhaps counter-intuitively, removing a face from the complex may induce \mathcal{A} to grow; unlike the case of standard persistence, we may have neither $\mathcal{A}_{i+1} \subseteq \mathcal{A}_i$ nor $\mathcal{A}_i \subseteq \mathcal{A}_{i+1}$.

Second, we consider the update of the (co)homology basis, defined on the critical faces $(\mathcal{A}_i, \tilde{\partial})$ of the Morse complex. The interesting case is when we have to remove a simplex $\sigma \in \mathcal{K}$, but not its paired face $\tau \in \mathcal{Q}$. Removing (τ, σ) from the matching will impact $\tilde{\partial}$, which would not have been the case if $\tau \in \Sigma$, and will temporarily make them both critical. Therefore the (co)homology basis requires an update from two levels: one from the new boundary operator and one from the addition of σ and τ in \mathcal{A}_i followed by the removal of σ . Both are directly related and executed at the same time.

Let n be the number of simplices added and removed in a zigzag filtration and $|K|$ the maximal number of simplices in any complex. Thanks to this approach, we obtain an algorithm to compute zigzag persistence whose complexity is reduced from $O(n|K|^2)$ to $O(n(|A|^2 + c))$, where $|A|$ is the maximal size of any Morse complex and c is the time complexity for computing and updating a Morse matching. In practice, $|A|$ is usually much smaller than $|K|$, and there exist very fast heuristics to compute and update Morse matchings, such that c is a small constant in practice.

References

- 1 G. E. Carlsson, V. de Silva, and D. Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the 25th ACM Symposium on Computational Geometry*, 2009.
- 2 R. Forman. A User's Guide to Discrete Morse Theory. In *Séminaire Lotharingien de Combinatoire*, volume 48, 2002.
- 3 C. Maria and S. Y. Oudot. Zigzag persistence via reflections and transpositions. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015.
- 4 K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2), 2013.

Stitch Fix for Mapper

Bala Krishnamoorthy¹

Department of Mathematics and Statistics, Washington State University

Nathaniel Saul¹

Department of Mathematics and Statistics, Washington State University

Bei Wang¹

School of Computing, Scientific Computing and Imaging Institute, University of Utah

1 Introduction

Mapper is one of the main tools in topological data analysis (TDA) and visualization used for the study of multivariate data [3]. It takes as input a multivariate function and produces a summary of high-dimensional data using a cover of the range space of the function. For a given cover, such a summary converts the mapping into a simplicial complex suitable for data exploration.

In this abstract, we take a *constructive* viewpoint of a multivariate function $f : \mathbb{X} \rightarrow \mathbb{R}^d$ and consider it as a vector of continuous, real-valued functions defined on a shared domain, $f = (f_1, f_2, \dots, f_d), f_i : \mathbb{X} \rightarrow \mathbb{R}$, where each f_i (referred to as a *filter function*) gives rise to a 1-dimensional mapper construction. We investigate a method for stitching a pair of mappers together and study a topological notion of information gain from such a process. In particular, we aim to assign a measure that captures information about how each filter function contributes to the topological complexity of the stitched result, and how the two filter functions are topologically correlated.

We are inspired by the ideas of stepwise regression for model selection and of scatterplot matrices for visualization. For a set of variables x_1, x_2, \dots, x_d , *stepwise regression* iteratively incorporates variables into a regression model based on some criterion. A measure of topological information gain can be used as a criterion for choosing filter functions, and to construct a single *best* mapper. The *scatterplot matrix* shows all pairwise scatterplots of the set of variables on a single $d \times d$ matrix. We introduce a topological analogue of the scatterplot matrix for a set of filter functions f_1, f_2, \dots, f_d , and study the degree of *topological correlation* between filter functions.

We define a composition (or stitching) operation for mappers (Definition 1) and show its equivalence to the standard construction (Theorem 2). We end by describing our ongoing effort in studying a topological notion of information gain and correlation between filter functions.

2 Preliminary Results

Given a space \mathbb{X} , a function $f : \mathbb{X} \rightarrow \mathbb{R}^d$, and a cover $\mathcal{U} = \{U_i\}$ of $f(\mathbb{X})$, we define the pullback cover $f^*(\mathcal{U})$ as the cover obtained by decomposing each $f^{-1}(U_i)$ into its path-connected components $U_i = \cup_{j=1}^{k_i} V_{ij}$. *Mapper* is then a simplicial complex defined as the nerve of this pullback cover $M(f, \mathcal{U}) := \text{Nrv}(f^*(\mathcal{U}))$.

► **Definition 1** (Composition). Given $f, g : \mathbb{X} \rightarrow \mathbb{R}$ and covers of their images, $\mathcal{U} = \{U_i\}, \mathcal{V} = \{V_j\}$, we construct a composed cover \mathcal{W} of \mathbb{X} from $f^*(\mathcal{U})$ and $g^*(\mathcal{V})$ by taking the connected

¹ Funding: NSF DBI-1661348 and NSF DBI-1661375

components of the following set, where $u \in f^*(\mathcal{U})$, $v \in g^*(\mathcal{V})$ are path-connected cover elements of \mathbb{X} ,

$$\{u \cap v \mid \forall u \in f^*(\mathcal{U}), \forall v \in g^*(\mathcal{V}), u \cap v \neq \emptyset\}.$$

We define the composed mapper as the nerve of this cover \mathcal{W} ,

$$S(M(f, \mathcal{U}), M(g, \mathcal{V})) := \text{Nrv}(\mathcal{W}).$$

Under certain assumptions, this composition S is equivalent to the classical method of constructing mappers from a pair of filter functions, as described by Theorem 2.

► **Theorem 2.** *If f and g are continuous real-valued functions, U_i, V_j , and $U_i \times V_j$ are simply connected for all i, j , then $S(M(f, \mathcal{U}), M(g, \mathcal{V})) = M((f, g), \mathcal{U} \times \mathcal{V})$, the mapper constructed in the traditional manner.*

Proof sketch. The proof follows directly from properties of continuous functions and connected sets. We provide a sketch here. Starting with the two covers associated with the two 1-dimensional mappers, \mathcal{U} for f and \mathcal{V} for g , we can show that the defined set \mathcal{W} and the cover obtained from the traditional mapper construction are equivalent. Taking the nerve of each, we can conclude that the resulting mapper are equivalent as well. ◀

Furthermore, we give an algorithm that illustrates how the composition can be considerably simplified by directly incorporating simplex information from each of the two input mappers. The algorithm that combines (or stitches) two mappers together works by tracking vertices (i.e. representatives of the pull back cover elements as a result of the Nrv operation) of the first mapper in a breadth first search fashion, and combining them with vertices of the second mapper. The simplices in both mappers provide hints about which possible simplices could be in the composition. Using this information to avoid many explicit intersection checks, we can considerably simplify and speedup the composition process. While some simplicies from each 1-dimensional mapper can be added directly to the composition (the **stitch** step), others require explicitly checking the nerve condition in the mapper construction (the **fix** step).

3 Discussion

As part of our ongoing research, we propose measures of information gain (i.e., the increase in topological complexity) from the composition process as well as topological correlations between pairs of filter functions. By tracking the **stitch** and **fix** steps of the construction process, it is possible to quantify the relationship between filter functions.

With such measures in hand, we return to our topological analogues of the stepwise regression [1] and scatter plot matrix [2], which help to navigate topological relationships among multiple filter functions. A method for *stepwise stitching* would produce a mapper with optimal topological information by iteratively building a multi-dimensional mapper from topologically independent filter functions. A *topological scatter plot matrix* can reveal information about the filter functions such as topological dependencies and outliers by providing a visualization of the most information rich filter functions.

References

- 1 M.A. Efroymson. Multiple regression analysis. In Anthony Ralston and Herbert S. Wilf, editors, *Mathematical Methods for Digital Computers*. Wiley, 1960.

- 2 Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.
- 3 Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. *Eurographics Symposium on Point-Based Graphics*, 22, 2007.

Curvature Estimates of Point Clouds as a Tool in Quantitative Prostate Cancer Classification

Anna Schenfisch¹

Department of Mathematical Sciences, Montana State University
Bozeman, MT, USA.

annaschenfisch@montana.edu

 <https://orcid.org/0000-0003-2546-5333>

Brittany Terese Fasy²

School of Computing and Dpt. of Mathematical Sciences, Montana State University
Bozeman, MT, USA.

brittany.fasy@montana.edu

 <https://orcid.org/0000-0003-1908-0154>

Prostate cancer is one of the most commonly diagnosed cancers. Diagnosis involves several factors, including analysis of 2D cross-sections of prostate tissue needle core biopsies. In such biopsies, glands appear as loops defined by the nuclei of cells defining the gland. As cancer progresses, glands often transform from circular to finger-like to unstructured, as molecules that keep the glands together are no longer sufficiently expressed in cancerous cells. Currently, the severity of cancer (and thus, treatment recommendations) is determined using the Gleason grading system, a visual analysis that compares biopsy features to a standard set of patterns in gland size, shape, and organization, and assigns regions of biopsies grades from 1 to 5 [4]. In particular, pathologists analyze the appearance the tubelike glands of such cross sections. Typically, a less cancerous prostate has fairly circular tubelike glands, whereas a more cancerous prostate has less uniformly circular glands; see Figure 1 for examples.

Although the Gleason grading system is certainly helpful to patients and doctors, its qualitative nature has the potential to lead to inconsistencies in scores given to biopsies [2] [5]. Such inconsistencies motivate the goal of quantifying gland curvature to aid in developing a more consistent method of classifying prostate cancer. In this paper, we propose a method to describe the shape of glands using curvature.

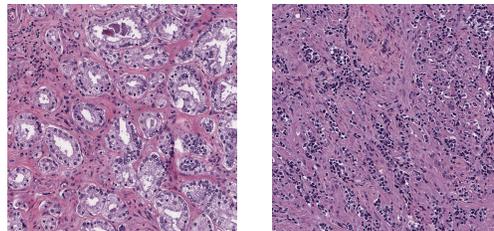


Figure 1 Example of stained cross sections of needle core biopsies of prostate tissue. The dark purple dots correspond to nuclei and outline each tubular gland. The example on the left contains fairly uniformly curved tubular glands, and would not be classified as severely cancerous. The example on the right shows a more cancerous sample; note that glands are beginning to lose their loop-like structure, and instead form sheets of cells. Our methods are particularly applicable to glands which still retain a clear loop structure.

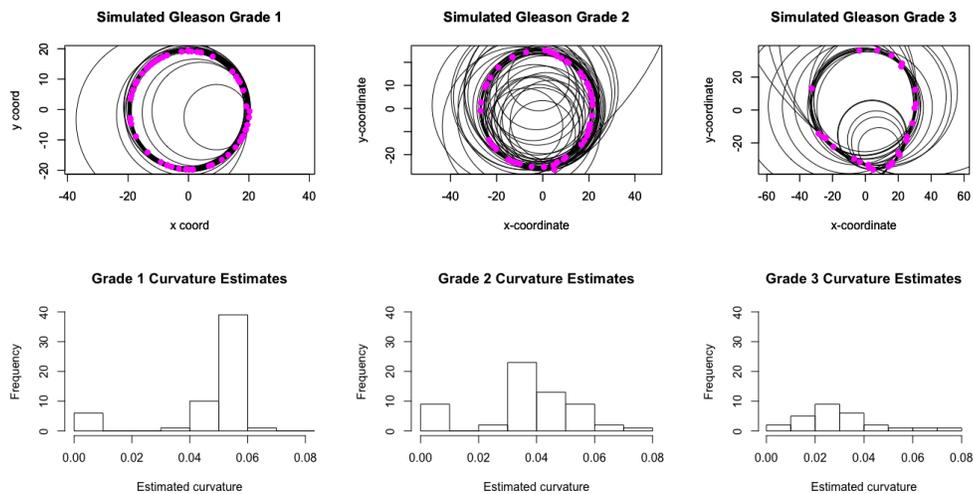
¹ NSF DMS 1664858

² NSF DMS 1664858

Methods We focus on studying the curvature of glands. To use this in practice, we would need to take the following steps: (1) extract nuclei defining a single gland; (2) order the nuclei in a counter-clockwise loop around the gland; (3) define curvature for a discretized curve. We focus on step (3). We defer step (1) to the full paper and use the CRUST algorithm [1] for step (2).

For a smooth curve C embedded in \mathbb{R}^2 , the extrinsic curvature at a point $x \in C$ is equal to $1/r$, where r is the radius of the circle that best approximates C at x [6]; the total curvature of a piecewise linear curve can be captured by turning angles [7]. We emulate the definition of curvature given above for smooth curves to estimate the curvature of prostate gland cross sections. For each nuclei n_i , we estimate the curvature by finding the best fitting circle containing n_i along with m neighbors on each side. Note that since a circle is defined by a minimum of three points, we require $m \geq 1$. Since we used the CRUST algorithm to find an ordering of our nuclei around a gland, these are the m nuclei before n_i and m nuclei after n_i . (If the number of nuclei is less than $2m + 1$, then we have duplicates in this set). Doing this for each nucleus, we obtain a distribution of local curvature estimates.

Experimental Results To test our method, we computed curvature distributions on simulated glands [3] for three different aggression levels; see Figure 2. As expected, preliminary results on simulated glands indicate that more cancerous glands tend to have higher variation in estimated local curvature than less cancerous glands.



■ **Figure 2** Curvature distributions for three simulated glands. The magenta dots in the top row are a simulation of the position of nuclei in glands. The best fitting circle for each nuclei using m neighbors on each side is shown (we used $m = 2$). Curvature at a nuclei is then estimated as the reciprocal such a circle, the corresponding histogram of which is shown on the bottom row.

Continued Research and Acknowledgements The next step is to study how the curvature distribution varies with Gleason grade and cancer severity for human biopsy data, which will ultimately be used in automated histology slide analysis.

The authors would like to thank Sawyer Payne for his work in developing gland simulation. Deidentified images in Figure 1 were obtained at Tulane University under an Institutional Review Board approved protocol.

References

- 1 Tamal K. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*, volume 23. Cambridge University Press, 2006.
- 2 Jonathan I. Epstein, Lars Egevad, Mahul B. Amin, Brett Delahunt, John R. Srigley, Peter A. Humphrey, Grading Committee, and others. The 2014 International Society of Urological Pathology (ISUP) consensus conference on Gleason grading of prostatic carcinoma: Definition of grading patterns and proposal for a new grading system. *The American journal of surgical pathology*, 40(2):244–252, 2016.
- 3 Brittany Terese Fasy, Sawyer Payne, Anna Schenfisch, Jordan Schupback, and Nathan Stouffer. Simulating prostate cancer slide scans. In preparation, 2018.
- 4 Peter A. Humphrey. Gleason grading and prognostic factors in carcinoma of the prostate. *Modern Pathology*, 17(3):292–306, 2004.
- 5 Bridget F Koontz, Matvey Tsivian, Vladimir Mouraviev, Leon Sun, Zeljko Vujaskovic, Judd Moul, and W Robert Lee. Impact of primary gleason grade on risk stratification for gleason score 7 prostate cancers. *International Journal of Radiation Oncology - Biology - Physics*, 82(1):200–203, 2012.
- 6 Shlomo Sternberg. *Curvature in Mathematics and Physics*. Courier Corporation, 2013.
- 7 John M. Sullivan. Curves of finite total curvature. In Alexander I. Bobenko, John M. Sullivan, Peter Schröder, and Günter M. Ziegler, editors, *Discrete Differential Geometry*, pages 137–161. Springer, 2008.

Maximum clique of disks in convex position

Onur Çağırıcı¹

Faculty of Informatics, Masaryk University Brno, Czech Republic
onur@mail.muni.cz

Bodhayan Roy

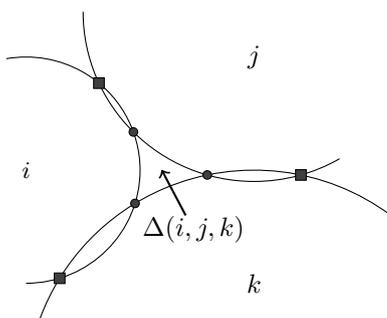
Faculty of Informatics, Masaryk University Brno, Czech Republic
b.roy@fi.muni.cz

Abstract

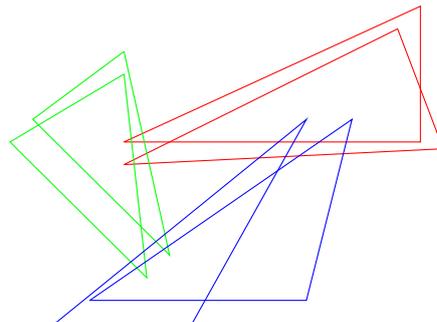
We consider the maximum clique problem on disk sets. Given a set \mathcal{D} of disks on the Euclidean plane, a maximum clique is a subset $\mathcal{Q} \subseteq \mathcal{D}$ where every pair $x, y \in \mathcal{Q}$ mutually intersect. Finding a maximum clique in a given disk set is polynomial time if the disks are unit disks, [4], but it is still an open problem even for disks with two different radii, [3], and naturally for the general case [1, 5]. We show that, if the disks are in a convex position, *i.e.*, if every disk in \mathcal{D} intersects the boundary of the convex hull of \mathcal{D} , then a clique in that set can be pierced by two points. This finding leads to a polynomial time algorithm via König's theorem, since the complement of this disk set indicates a bipartite graph.

We write “a point pierces a disk” if the mentioned disk contains that point. If two disks intersect, and their common intersection region is intersected by no other disks, then that region is referred to as a *lens*. We refer to a common intersection region among multiple disks as a *cell*. If a given cell is not intersected by any disk, then it is a *minimal cell*. When three disks i , j , and k are mutually intersecting, and their lenses are mutually disjoint, we say i , j and k are *intersecting privately*. If there are no triplets that intersect privately, then up to Helly property, the clique can be pierced by a single point [2].

Suppose that three disks, i, j, k , are intersecting privately. There exists an empty region enclosed by these three disks and also not contained in any of the three disks. We call this region a *concave triangle*, and denote it by $\Delta(i, j, k)$. We name the corners of a concave triangle as the *inner intersection points*. The remaining points are the *outer intersection points*. The inner intersection points are denoted by dots, and the outer intersection points are denoted by rectangles in Figure 1a.



(a) Three privately intersecting disks, their intersection points, and the concave triangle in the middle.



(b) A clique of six triangles that cannot be pierced by two points; because every pair of triangles with the same color is involved in two private intersections with a pair of another color.

■ **Figure 1** (a) intersection points (b) a clique of six triangles

¹ O. Çağırıcı is supported by the Czech Science Foundation, project no. 17-00837S.

21:2 Maximum clique of disks in convex position

► **Theorem 1** (Main theorem). *If a set of pairwise intersecting disks are in a convex position, then they can be pierced by two points.*

To prove Theorem 1, we consider the concave triangle with the maximum area.

► **Lemma 2.** *Given three disks $i, j, k \in \mathcal{Q}$ where the concave triangle $\Delta(i, j, k)$ has maximum area, every other disk $d \in \mathcal{Q}$ contains at least one of the inner intersection points obtained by the intersections of the disks i, j and k .*

Let \mathcal{Q} be the maximum clique of a disk graph, and let $i, j, k \in \mathcal{Q}$ be three privately intersecting disks that form a concave triangle with maximum area. We call these disks as the *main disks*, the lenses between each pair as the *main lenses*, and the concave triangle with maximum area as the *main triangle*.

► **Lemma 3.** *If a disk d contains exactly one inner intersection point, then it contains the corresponding outer intersection point.*

► **Corollary 4.** *Every disk $d \in \mathcal{Q}$ contains one of the main lenses.*

Without loss of generality, let i, j, k be the main disks with centers in clockwise order, and let \mathbb{A}, \mathbb{B} and \mathbb{C} be the main lenses in clockwise order, beginning from the rightmost main lens. That is, \mathbb{A} is the lens formed by i and j , \mathbb{B} is the lens formed by j and k , and \mathbb{C} is the lens formed by i and k . If there are no three disks except the main disks intersect privately, then we can pick the two piercing points as follows. Considering three minimal cells inside each main lens, pick any point in one of them, and any point in another.

For the other case, let us define the subsets A, B and C of \mathcal{D} . If a disk d contains \mathbb{A} , then $d \in A$. Else if d does not contain \mathbb{A} but contains \mathbb{B} , then $d \in B$. Else if d does contain neither \mathbb{A} nor \mathbb{B} but contains \mathbb{C} , then $d \in C$. Furthermore, we define three subsets for each of these sets using the following naming convention. If a disk $d \in A$ intersects \mathbb{B} , then $d \in A_B$. Else if $d \in A$ does not intersect \mathbb{B} but intersects \mathbb{C} , then $d \in A_C$. Else if $d \in A$ does intersect neither of \mathbb{B} or \mathbb{C} , then $d \in A'$.

In the following lemma we identify a crucial property which is very unique to disks. See Figure 1b for a (counter)example with convex sets in place of disks.

► **Lemma 5.** *Consider three pairs $(a, b), (u, v)$ and (x, y) of disks in \mathcal{Q} where the disks in each pair intersect. Suppose that $(a \cap u) \cap (b \cap u) = \emptyset$, and $(a \cap v) \cap (b \cap v) = \emptyset$. Also suppose that $(u \cap x) \cap (v \cap x) = \emptyset$, and $(u \cap y) \cap (v \cap y) = \emptyset$. Then either $(x \cap a) \cap (y \cap a) \neq \emptyset$ or $(x \cap b) \cap (y \cap b) \neq \emptyset$ holds.*

Let ϕ and ψ be two piercing points. We pick ϕ inside a minimal cell in one of the main lenses, say \mathbb{A} . Then, by Corollary 4, ϕ pierces the disks in $A \cup B_A \cup C_A \cup i \cup j$, and thus it remains to show that all disks in $B' \cup B_C \cup C' \cup C_B \cup k$ have a common intersection region. To show this, we first show that ϕ pierces all the private intersections in \mathbb{A} . Suppose $b_1, b_2, \dots, b_m \in B$ have private intersections with $c_1, c_2, \dots, c_p \in C$ inside the lens \mathbb{A} . Without loss of generality, let no pair of b_1, \dots, b_m have intersection inside \mathbb{A} . That is, they create “bulges” from one side, and every c_1, \dots, c_p intersects these bulges from the other side. When we pierce the minimal cell inside \mathbb{A} , we also pierce the disks c_1, \dots, c_p .

For the remaining disks, they cannot intersect outside of k because B' intersects i and C' intersects j , and thus have a part of their lens in k . By definition, B_C intersects i only at \mathbb{C} , and C_B intersects j only at \mathbb{B} . This causes them to intersect all the disks in B' and C' inside k . As a result, ψ is a point in the minimal cell inside k .

► **Corollary 6.** *Given a set of disks in convex position, the maximum clique problem on that set can be computed in polynomial time.*

References

- 1 C. Ambühl and U. Wagner. The Clique Problem in Intersection Graphs of Ellipses and Triangles. *Theory of Computing Systems*, 38(3):279–292, May 2005.
- 2 B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability*. Cambridge University Press, New York, NY, USA, 1986.
- 3 S. Cabello. Maximum clique for disks of two sizes. <http://cgweek15.tcs.uj.edu.pl/problems.pdf>. Accessed: 5th June 2018.
- 4 B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1):165 – 177, 1990.
- 5 A. V. Fishkin. Disk Graphs: A Short Survey. In *Approximation and Online Algorithms*, pages 260–264. Springer Berlin Heidelberg, 2004.

2012 ACM Subject Classification G.2.2 Graph Theory, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Computational geometry, disk intersection, maximum clique

An Algorithmic Framework for Geometric Intersection Graphs^{*†}

Mark de Berg

Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Hans L. Bodlaender

Dept. of Computer Science, Utrecht University and Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Sándor Kisfaludi-Bak

Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Dániel Marx

Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary

Tom C. van der Zanden

Department of Computer Science, Utrecht University, The Netherlands

Many hard graph problems that seem to require $2^{\Omega(n)}$ time on general graphs, where n is the number of vertices, can be solved in subexponential time on planar graphs. In particular, many of these problems can be solved in $2^{O(\sqrt{n})}$ time on planar graphs, e.g. INDEPENDENT SET, DOMINATING SET, and HAMILTONIAN CYCLE. This so-called *square-root phenomenon* [14] is a consequence of the planar separator theorem [12, 13] and treewidth-based algorithms [6]. These methods give a general framework to obtain subexponential algorithms on planar graphs or, more generally, on H -minor free graphs. It builds heavily on the fact that H -minor free graphs have treewidth $O(\sqrt{n})$ and, hence, admit a separator of size $O(\sqrt{n})$. A similar line of work is emerging in the area of geometric intersection graphs, with running times that have $O(n^{1-1/d})$ in the exponent in the d -dimensional case [16, 17], where $d \geq 2$. Our goal is to establish a framework for a wide class of geometric intersection graphs that is similar to the framework known for planar graphs.

The *intersection graph* $G[F]$ of a set F of objects in \mathbb{R}^d is the graph whose vertex set is F and in which two vertices are connected when the corresponding objects intersect. (*Unit-disk graphs*, where F consists of (unit) disks in the plane are a widely studied class of intersection graphs. They form a natural generalization of planar graphs, since any planar graph can be realized as the intersection graph of a set of disks in the plane. In this paper we consider intersection graphs of a set F of *fat objects*, where we call an object $o \subseteq \mathbb{R}^d$ is α -*fat*, for some $0 < \alpha \leq 1$ if there are balls B_{in} and B_{out} in \mathbb{R}^d such that $B_{\text{in}} \subseteq o \subseteq B_{\text{out}}$ and $\text{radius}(B_{\text{in}})/\text{radius}(B_{\text{out}}) \geq \alpha$. For example, disks are 1-fat and squares are $(1/\sqrt{2})$ -fat. From now on we assume that α is an absolute constant, and often simply speak of fat objects. Note that we do not require the objects in F to be convex, or even connected. Thus our definition is very general. In particular, it does not imply that F has near-linear union complexity, as is the case for so-called locally-fat objects [2]. In most of our results we furthermore assume that the objects in F are *similarly sized*, meaning that the ratio of their diameters is bounded by a fixed constant. Several important graph problems have

* This work was supported by the NETWORKS project, funded by the Netherlands Organization for Scientific Research NWO under project no. 024.002.003 and by the ERC Consolidator Grant SYSTEMATICGRAPH (No. 725978) of the European Research Council.

† A longer version of this work will appear in the proceedings of the 50th Annual ACM Symposium on the Theory of Computing (STOC 2018). A preprint is available at [7].

been investigated for (unit-)disk graphs or other types of intersection graphs [1, 3, 8, 9, 16]. However, an overarching framework that helps designing subexponential algorithms has remained elusive. A major goal of our work is to give a framework that can even be applied when the ply (the maximum number of objects intersecting in any fixed point) is unbounded.

Our contribution. As mentioned, many subexponential results for planar graphs rely on planar separators. Our first contribution is a generalization of this result to intersection graphs of (arbitrarily-sized) fat objects in \mathbb{R}^d . Since these graphs can have large cliques we cannot bound the number of vertices in the separator. Instead, we build a separator consisting of cliques. We then define a weight function γ on these cliques—in our applications it suffices to define the weight of a clique C as $\gamma(|C|) := \log(|C| + 1)$. We define the weight of a separator as the sum of the weights of its constituent cliques C_i , which is useful since for many problems a separator can intersect the solution vertex set in $2^{O(\sum_i \gamma(|C_i|))}$ many ways. The theorem states that given a set of fat objects in some fixed constant dimension d , a clique decomposition and separator of weight $O(n^{1-1/d})$ exists, and it can be found in polynomial time if the objects are “nice”. A direct application of our separator theorem is a $2^{O(n^{1-1/d})}$ algorithm for INDEPENDENT SET. For general fat objects, only the 2-dimensional case was known to have such an algorithm [15].

After proving the weighted separator theorem for arbitrarily-sized fat objects, we switch to similarly-sized objects. A desirable property of algorithms for geometric graphs is that they are *robust*, meaning that they can work directly on the graph without knowledge of the underlying geometry. Most of the known algorithms are in fact non-robust, which could be a problem in applications, since finding a geometric representation of a given geometric intersection graph is NP-hard [5] (and many recognition problems for geometric graphs are ER-complete [11]). One of the advantages of our framework is that it yields robust algorithms for many problems. Here the idea is as follows: We find a certain partition \mathcal{P} of the intersection graph G , with the property that each class is connected and the union of $\kappa = O(1)$ cliques (such partitions are called κ -partitions). We contract each class to a single vertex, and then work with the contracted graph $G_{\mathcal{P}}$ where the node corresponding to a class C gets weight $\gamma(|C|)$. We show that such a partition can be found efficiently without knowing the set F defining the given intersection graph.

► **Theorem 1.** *Let $G = (V, E)$ be the intersection graph of an (unknown) set of n similarly-sized α -fat objects in \mathbb{R}^d , and let γ be a weight function such that $1 \leq \gamma(t) = O(t^{1-1/d-\epsilon})$, for constants $d \geq 2$, $\alpha > 0$, and $\epsilon > 0$. Then there exist constants κ and Δ such that there is a κ -partition \mathcal{P} with the following properties: (i) $G_{\mathcal{P}}$ has maximum degree at most Δ , and (ii) $G_{\mathcal{P}}$ has weighted treewidth $O(n^{1-1/d})$. Moreover, such a partition \mathcal{P} and a corresponding weighted tree decomposition of width $O(n^{1-1/d})$ can be computed in $2^{O(n^{1-1/d})}$ time.*

Furthermore, we show that our approach can be combined with the *rank-based approach* [4], a technique to speed up algorithms for connectivity problems. Thus we obtain a framework that gives $2^{O(n^{1-1/d})}$ -time algorithms for intersection graphs of similarly-sized fat objects for many problems for which treewidth-based algorithms are known. Our framework recovers and often slightly improves the best known results for several problems, including INDEPENDENT SET, HAMILTONIAN CYCLE and FEEDBACK VERTEX SET. Our framework also gives the first subexponential algorithms in geometric intersection graphs for, among other problems, r -DOMINATING SET for constant r , STEINER TREE and CONNECTED DOMINATING SET. We obtain robust algorithms for many of the problems mentioned above, in contrast to known results which almost all need the underlying set F as input. Note that we have matching upper and lower bounds, so we show that the best possible running time for these problems is $2^{\Theta(n^{1-1/d})}$, unless the Exponential Time Hypothesis [10] fails.

References

- 1 Jochen Alber and Jirí Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004. URL: <https://doi.org/10.1016/j.jalgor.2003.10.001>, doi:10.1016/j.jalgor.2003.10.001.
- 2 Boris Aronov, Mark de Berg, Esther Ezra, and Micha Sharir. Improved bounds for the union of locally fat objects in the plane. *SIAM Journal on Computing*, 43(2):543–572, 2014. URL: <https://doi.org/10.1137/120891241>, doi:10.1137/120891241.
- 3 Csaba Biró, Édouard Bonnet, Dániel Marx, Tillmann Miltzow, and Paweł Rzażewski. Fine-grained complexity of coloring unit disks and balls. In *Proceedings of the 33rd International Symposium on Computational Geometry, SoCG 2017*, volume 77 of *LIPCS*, pages 18:1–18:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. URL: <https://doi.org/10.4230/LIPICS.SoCG.2017.18>, doi:10.4230/LIPICS.SoCG.2017.18.
- 4 Hans L Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015.
- 5 Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Comput. Geom.*, 9(1-2):3–24, 1998. URL: [https://doi.org/10.1016/S0925-7721\(97\)00014-X](https://doi.org/10.1016/S0925-7721(97)00014-X), doi:10.1016/S0925-7721(97)00014-X.
- 6 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 7 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for ETH-tight algorithms and lower bounds in geometric intersection graphs. *CoRR*, abs/1803.10633, 2018. URL: <http://arxiv.org/abs/1803.10633>, arXiv:1803.10633.
- 8 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPICS*, pages 65:1–65:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. URL: <https://doi.org/10.4230/LIPICS.ICALP.2017.65>, doi:10.4230/LIPICS.ICALP.2017.65.
- 9 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012*, pages 1563–1575. SIAM, 2012. URL: <http://portal.acm.org/citation.cfm?id=2095240&CFID=63838676&CFTOKEN=79617016>.
- 10 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k -SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. URL: <https://doi.org/10.1006/jcss.2000.1727>, doi:10.1006/jcss.2000.1727.
- 11 Ross J. Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete & Computational Geometry*, 47(3):548–568, 2012. URL: <https://doi.org/10.1007/s00454-012-9394-8>, doi:10.1007/s00454-012-9394-8.
- 12 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
- 13 Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.
- 14 Dániel Marx. The square root phenomenon in planar graphs. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, page 28, 2013. URL: https://doi.org/10.1007/978-3-642-39212-2_4, doi:10.1007/978-3-642-39212-2_4.

- 15 Dániel Marx and Michal Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *Proceedings of the 23rd Annual European Symposium on Algorithms, ESA 2015*, volume 9294 of *Lecture Notes in Computer Science*, pages 865–877. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-48350-3_72, doi:10.1007/978-3-662-48350-3_72.
- 16 Dániel Marx and Anastasios Sidiropoulos. The limited blessing of low dimensionality: when $1 - 1/d$ is the best possible exponent for d -dimensional geometric problems. In *Proceedings of the 30th Annual Symposium on Computational Geometry, SOCG 2014*, pages 67–76. ACM, 2014. URL: <http://doi.acm.org/10.1145/2582112.2582124>, doi:10.1145/2582112.2582124.
- 17 Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science, FOCS 1998*, pages 232–243. IEEE Computer Society, 1998. URL: <https://doi.org/10.1109/SFCS.1998.743449>, doi:10.1109/SFCS.1998.743449.

Conical partitions of point sets

Gábor Damásdi¹

The Hebrew University of Jerusalem, Israel
gabor.damasdi@mail.huji.ac.il

Abstract

Mass partition theorems have been extensively studied in recent decades. Conical partitions have been mainly considered in the planar case [1, 2, 4], but some higher dimensional results have been obtained by Vrećica and Živaljević [5] and Makeev [3].

The proof of mass partition theorems usually follows the configuration space/test map procedure or some degree theoretic method, both of which heavily rely on topological results. This is especially true for results in higher dimensions, where our combinatorial tools are limited. We show a completely combinatorial proof for a discrete version of a theorem of Vrećica and Živaljević [5] concerning conical partitions.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems

Keywords and phrases mass partition, conical dissection, combinatorial proof

1 Introduction

A *pointed simplex* is a pair consisting of a nondegenerate simplex in \mathbb{R}^d and a point in its interior. For a simplex $\Delta = \text{conv}(a_0, a_1, \dots, a_d)$ that contains $\mathbf{0}$ in its interior let (Δ, a) denote the pointed simplex $\Delta + a$ with a chosen as the interior point. Let $R_i(\Delta, a)$ be the cone with apex at a and generating vectors $(a_0, \dots, a_{i-1}, a_{i+1}, \dots, a_d)$. In other words, $R_i(\Delta, a)$ is the cone that is spanned by the i -th face of the simplex Δ and translated to have apex at a . The dissection of \mathbb{R}^d into cones $R_0(\Delta, a), \dots, R_d(\Delta, a)$ is a *radial (Δ, a) -dissection* and denoted by $R_{(\Delta, a)}$.

Let S be a set of n points in \mathbb{R}^d and let $\alpha = (k_0, k_1, \dots, k_d)$ be a nonnegative integer vector such that $k_0 + k_1 + \dots + k_d = n$. Then $R_{(\Delta, a)}$ is called an α -*partition* if

$$|S \cap \text{int}(R_i(\Delta, a))| = k_i, \quad i = 0, 1, \dots, d$$

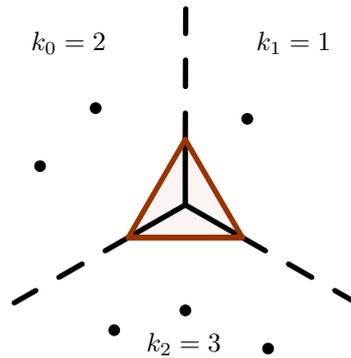
We show that any α -partition can be realized by a suitable translate of any given pointed simplex.

► **Theorem 1.** *Let S be a set of n points in \mathbb{R}^d and let $\alpha = (k_0, k_1, \dots, k_d)$ be a nonnegative integer vector such that $k_0 + k_1 + \dots + k_d = n$. Let $(\Delta, \mathbf{0})$ be a pointed simplex such that S is in general position with respect to Δ . Then there exists a vector $v \in \mathbb{R}^d$ such that the radial dissection $R_{(\Delta, v)}$ is an α -partition of S .*

Here *general position with respect to Δ* means that certain finitely many affine subspaces that depend on Δ and S don't admit any nontrivial intersection.

Vrećica and Živaljević considered α -partitions for admissible measures. Theorem 1 can be viewed as a discrete version of Theorem 2.4 in [5]. Their proof follows the ideas of the Brouwer fixed-point theorem and they argue that the configuration space/test map procedure cannot be applied to this theorem. We give an elementary proof of Theorem 1, which we outline here.

¹ G. Damásdi was supported by ISF grant 1050/16 and NKFIH grant NKFI-K119670



■ Figure 1

Notice that $p \in R_i(\Delta, q)$ if and only if $q \in R_i(-\Delta, p)$. So it is useful to consider the dissection of \mathbb{R}^d into regions by the cones $R_i(-\Delta, p)$ ($i = 0, 1, \dots, d, p \in S$). Clearly two dissection $R_{(\Delta, v_1)}$ and $R_{(\Delta, v_2)}$ induce the same partition of S if and only if v_1 and v_2 lie in the same region. We show something stronger, that two dissection $R_{(\Delta, v_1)}$ and $R_{(\Delta, v_2)}$ partition S to subsets of the same size if and only if v_1 and v_2 lie in the same region.

Hence it is enough to show that there are as many regions as possible α vectors. An α vector contains $d + 1$ nonnegative integers that sum up to n , so there are $\binom{n+d}{d}$ of them. Therefore Theorem 1 is equivalent to the following statement.

► **Theorem 2.** *Let S be a set of n points in \mathbb{R}^d and let $(\Delta, 0)$ be a pointed simplex such that S is in general position with respect to Δ . Then the cones $R_i(-\Delta, p)$ ($i = 0, \dots, d, p \in S$) dissect \mathbb{R}^d into $\binom{n+d}{d}$ regions.*

We prove Theorem 1 and 2 together using induction on d . Since they are equivalent for each d and n we can interchangeably prove one of them from the lower dimensional cases. When we show Theorem 2 we use the following idea.

To count the regions we can pick an appropriate direction v as upward direction and consider the highest point in each region. Some regions don't have a highest point, we show by induction that there are $\binom{n+d-1}{d-1}$ of these. The rest of the regions are in one to one correspondence with their highest point, hence it is enough to count the points that appear as highest point of some region. We will see that the number of these points is $\binom{n+d-1}{d}$, therefore we have $\binom{n+d-1}{d-1} + \binom{n+d-1}{d} = \binom{n+d}{d}$ regions. ◀

References

- 1 I. Bárány and J. Matousek. Simultaneous partitions of measures by k -fans. *Discrete & Computational Geometry*, 25(3):317–334, 2001.
- 2 Imre Bárány, Pavle Blagojević, and András Szűcs. Equipartitioning by a convex 3-fan. *Advances in Mathematics*, 223(2):579–593, 2010.
- 3 V. V. Makeev. Equipartition of a mass continuously distributed on a sphere or in space. *Journal of Mathematical Sciences*, 119(1):96–100, 2004.
- 4 Toshinori Sakai. Balanced Convex Partitions of Measures in \mathbb{R}^2 . *Graphs and Combinatorics*, 18(1):169–192, 2002.
- 5 S. T. Vrecica and R. T. Zivaljević. Conical equipartitions of mass distributions. *Discrete & Computational Geometry*, 25(3):335–350, 2001.

Convex Partial Transversals of Planar Regions

Vahideh Keikha², Mees van de Kerkhof¹, Marc van Kreveld¹, Irina Kostitsyna³, Maarten Löffler¹, Frank Staals¹, Jérôme Urhausen¹, Jordi L. Vermeulen¹, and Lionov Wiratma¹

- 1 Department of Information and Computing Sciences, Utrecht University
`{m.a.vandekerkhof;m.j.vankreveld;m.loffler;f.staals;j.e.urhausen;
j.l.vermeulen;l.wiratma}@uu.nl`
- 2 Amirkabir University
`va.keikha@aut.ac.ir`
- 3 TU Eindhoven
`i.kostistyna@tue.nl`

1 Introduction

Let $\mathcal{R} = \{r_1, \dots, r_n\}$ be a set of n regions in the plane. We say that a set of points Q is a *partial transversal* of \mathcal{R} if there exists an injective map $f : Q \rightarrow \mathcal{R}$ such that $q \in f(q)$ for all $q \in Q$. For reasons of brevity we refer to partial transversals simply as “transversals” in the following. A transversal Q of \mathcal{R} is *convex* if Q is in convex position. We say that $Q = \{v_1, \dots, v_k\}$ is an *upper transversal* of \mathcal{R} if the chain v_1, \dots, v_k is x -monotone, convex and unimodal. In this paper, we study the problem of finding a convex partial transversal Q of a given cardinality $|Q| = k$ in a given set of regions \mathcal{R} .

This paper is an extension of the publications by Arkin *et al.* [1] and Schlipf [2]. We were able to prove that the optimization problem, where we try to find the transversal of maximum cardinality, is NP-hard even when the regions in \mathcal{R} do not intersect. Similarly, it is NP-hard to determine a maximal convex transversal when \mathcal{R} contains 3-oriented intersecting segments. For parallel segments we can determine a maximal upper convex transversal in $O(n^2)$ time and a maximal convex transversal in $O(n^6)$ time.

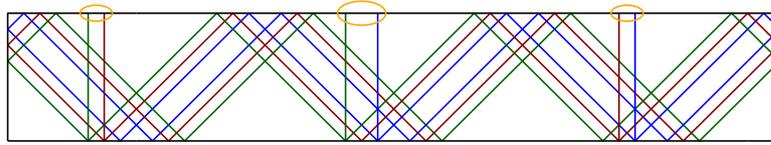
2 Parallel disjoint line segments

We restrict \mathcal{R} to a set of parallel line segments. We first search for an upper convex transversal.

► **Lemma 1.** *If there exists an upper convex transversal of \mathcal{R} that visits k regions, then there exists an upper convex transversal that visits the same regions, and that has the left- and rightmost vertex, and all strictly convex vertices, on the bottom endpoints of the regions in \mathcal{R} .*

We can therefore use a dynamic program to find the maximal upper convex transversal by processing the lower endpoints from left to right. For each lower endpoint v we determine for each previous lower endpoint u the size $s(u, v)$ of the maximal convex upper transversal that ends with u and v . The value of $s(u, v)$ is the sum of the number of regions intersected by the segment \overline{uv} and the largest $s(x, u)$ over all x left of u that are below the line uv . This gives us an $O(n^2)$ algorithm for determining the maximal upper convex transversal ending in v . We can reduce this to linear time using the point-line duality and by processing the vertices to the left of v in radially sorted order. This lets us determine the maximal upper convex transversal in $O(n^2)$ time. As we show in the full version, this approach can be adapted to work for the maximal convex transversal.

► **Theorem 2.** *Let \mathcal{R} be a set of parallel line segments. We can compute the maximal upper convex transversal Q of \mathcal{R} in $O(n^2)$ time and the maximal convex transversal in $O(n^6)$ time.*



■ **Figure 1** The construction for the MAX-2-SAT reduction. Each chain represents one variable. There is a gadget at each bounce of a chain and a clause gadget at the areas marked in orange.

3 3-oriented intersecting segments

► **Theorem 3.** *Let \mathcal{R} be a set of segments that have three different orientations. The problem of finding the maximal convex transversal Q of \mathcal{R} is NP-hard.*

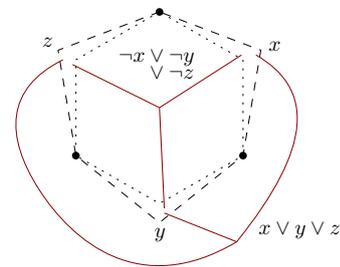
We take an instance of MAX-2-SAT (V, C) , where V is the set of variables and C is the set of clauses. We take an axis-aligned 1 by $|C|$ rectangle. For each variable v we create a chain c_v of diagonal segments bouncing inside the rectangle, as shown in Figure 1, such that the chains are close to each other. For each clause $x \vee y$ we create two vertical segments s_x and s_y close together, where s_v has its lower endpoint on one of the bounces of the chain c_v . The set of regions \mathcal{R} contains the segments of the chains, the clause segments and the gadgets described below.

Let Q be a convex transversal of \mathcal{R} , v be a variable and s be a segment within the chain c_v . For the point $q \in Q$ with $f(q) = s$, q is either the left or right endpoint of s . This determines the state of the variable (true/false). We construct gadgets at each bounce of a chain such that the chain has the same state everywhere. We then construct a clause gadget such that the convex transversal contains one more point if at least one of the chains is in the correct state.

4 Non-intersecting non-convex regions

► **Theorem 4.** *Finding the largest convex k -gon in disjoint non-convex regions is NP-hard.*

We reduce from MONOTONE PLANAR 3-SAT using the construction seen in Figure 2. Intuitively, we wrap the planar embedding of the 3-SAT instance around in a circle. The regions in our input are the vertices shared by the dotted and dashed polygons in the figure and the red regions representing the clauses. We need to place the regions such that there is no convex path between two subsequent vertices that intersect regions on both the “true” and “false” side. For n variables and m clauses, our 3-SAT instance is satisfiable if and only if there exists a convex transversal of cardinality $n + m$.



■ **Figure 2** Our construction for the reduction from MONOTONE PLANAR 3-SAT to finding the largest convex k -gon in disjoint non-convex regions.

References

- 1 Esther M. Arkin, Claudia Dieckmann, Christian Knauer, Joseph S.B. Mitchell, Valentin Polishchuk, Lena Schlipf, and Shang Yang. Convex transversals. In *Workshop on Algorithms and Data Structures*, pages 49–60. Springer, 2011.
- 2 Lena Schlipf. Notes on convex transversals. *arXiv preprint arXiv:1211.5107*, 2012.

Large equilateral sets in subspaces of ℓ_∞^n

Nóra Frankl¹

Department of Mathematics, London School of Economics and Political Science
Houghton Street, London WC2A 2AE, United Kingdom
n.frankl@lse.ac.uk

Abstract

For fixed k we prove exponential lower bounds on the equilateral number of subspaces of ℓ_∞^n of codimension k . In particular, we show that if the unit ball of a normed space of dimension n is a centrally symmetric polytope with at most $\frac{4n}{3} - o(n)$ pairs of facets, then it has an equilateral set of cardinality at least $n + 1$. Partially solving a problem of Kobos, we also prove that the equilateral number of subspaces of codimension 2 of ℓ_∞^n for $n \geq 9$ is at least $n - 1$.

2012 ACM Subject Classification

Keywords and phrases equilateral sets

Acknowledgements We thank Konrad Swanepoel for bringing our attention to Corollary 3 and for discussions on the topic.

1 Introduction and results

Let $(X, \|\cdot\|)$ be a normed space. A set $S \subseteq X$ is called *c-equilateral* if $\|x - y\| = c$ for all distinct $x, y \in S$. S is called *equilateral* if it is *c-equilateral* for some $c > 0$. The *equilateral number* $e(X)$ of X is the cardinality of the largest equilateral set of X . Petty [4] made the following conjecture regarding lower bounds on $e(X)$.

► **Conjecture 1** (Petty[4]). *For all normed spaces X of dimension n , $e(X) \geq n + 1$.*

Conjecture 1, except for some special class of norms, is still open for $n \geq 5$, with the best general lower bound $e(X) \geq \exp(\Omega(\sqrt{\log n}))$, proved by Swanepoel and Villa [5].

The norm $\|\cdot\|_\infty$ of $x \in \mathbb{R}^n$ is defined as $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$, and ℓ_∞^n denotes the normed space $(\mathbb{R}^n, \|\cdot\|_\infty)$. In [3] Kobos studied subspaces of ℓ_∞^n of codimension 1, and proved the lower bound $e(X) \geq 2^{\lfloor \frac{n}{2} \rfloor}$. He also proposed as a problem to prove Petty's conjecture for subspaces of ℓ_∞^n of codimension 2. In Theorem 2 we prove exponential lower bounds on the equilateral number of k -codimensional subspaces of ℓ_∞^n . This, in particular, solves Kobos' problem if $n \geq 9$.

As $e(X)$ is upper semi-continuous on the Banach-Mazur compactum and any convex polytope can be obtained as a section of a cube of sufficiently large dimension (see for example Page 72 of Grünbaum's book [2]), it would be sufficient to prove Conjecture 1 for k -codimensional subspaces of ℓ_∞^n for all $1 \leq k \leq n$. (This was also pointed out in [3].) Unfortunately, our bounds are only non-trivial if n is sufficiently large compared to k . However, we deduce an interesting corollary.

► **Theorem 2.** *Let X be a $(n - k)$ -dimensional subspace of ℓ_∞^n . Then the following inequalities hold.*

¹ Research was partially supported by the National Research, Development, and Innovation Office, NKFIH Grant K119670.

1. $e(X) \geq 1 + \max_{1 \leq \ell \leq \frac{n}{k+1}} \frac{1}{2^{k-1}} \sum_{1 \leq r \leq \ell} \binom{n-k\ell}{r}$.
2. $e(X) \geq 1 + \max_{1 \leq \ell \leq \frac{n}{2k+1}} \sum_{1 \leq r \leq \ell} \binom{n-2k\ell}{r}$.
3. $e(X) \geq \frac{2^{n-k}}{(n-k)^k}$.

Idea of proof of Theorem 2. For two vectors $\mathbf{x}, \mathbf{y} \in X$ we have $\|\mathbf{x} - \mathbf{y}\|_\infty = c$ if and only if the following hold.

$$\text{There is an } 1 \leq i \leq n \text{ such that } |x^i - y^i| = c. \quad (1)$$

$$|x^i - y^i| \leq c \text{ for all } 1 \leq i \leq n. \quad (2)$$

In our constructions of a 1-equilateral set $S \subseteq X$ we split the index set $[n]$ of the coordinates into two parts $[n] = N_1 \cup N_2$. In the first part N_1 , we choose all the coordinates from the set $\{0, 1, -1\}$, so that for each pair from S there will be an index in N_1 for which 1 holds, and 2 is not violated by any index in N_1 . We use the second part of the indices to ensure that all of the points we choose are indeed in the subspace X . For each vector, this will lead to a system of linear equations. The main difficulty will be to choose the values of the coordinates in N_1 so that the coordinates in N_2 , obtained as a solution to those systems of linear equations, do not violate 2. Choosing the coordinates in N_1 in different ways, will lead to different constructions with different lower bounds. We remark that the three inequalities are not redundant, none of them follows from the other two. \blacktriangleleft

► **Corollary 3.** *Let P an origin-symmetric convex polytope in \mathbb{R}^n with at most $\frac{4n}{3} - \frac{1+\sqrt{8n+9}}{6} = \frac{4n}{3} - o(n)$ pairs of facets. If X is a n -dimensional normed space with P as a unit ball, then $e(X) \geq n + 1$.*

We also prove a result regarding extensions of lower bounds to normed spaces that are close to subspaces of ℓ_∞^n according to the Banach-Mazur distance. The proof is based on using the Brouwer Fixed-Point Theorem, first applied in this context by Brass [1].

► **Theorem 4.** *Let X be an $(n-k)$ -dimensional subspace of ℓ_∞^n , and $d_{BM}(X, Y) \leq 1 + \frac{\ell}{2(n-2k-\ell k-1)}$ for some integer $1 \leq \ell \leq \frac{n-2k}{k}$. Then $e(Y) \geq n - k(2 + \ell)$.*

References

- 1 Peter Braß. On equilateral simplices in normed spaces. *Beiträge Algebra Geom.*, 40(2):303–307, 1999.
- 2 Branko Grünbaum. *Convex polytopes*, volume 221 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2003. Prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler. URL: <http://dx.doi.org/10.1007/978-1-4613-0019-9>, doi:10.1007/978-1-4613-0019-9.
- 3 Tomasz Kobos. Equilateral dimension of certain classes of normed spaces. *Numer. Funct. Anal. Optim.*, 35(10):1340–1358, 2014. URL: <http://dx.doi.org/10.1080/01630563.2014.930482>, doi:10.1080/01630563.2014.930482.
- 4 C. M. Petty. Equilateral sets in Minkowski spaces. *Proc. Amer. Math. Soc.*, 29:369–374, 1971. URL: <http://dx.doi.org/10.2307/2038145>, doi:10.2307/2038145.
- 5 Konrad J. Swanepoel and Rafael Villa. A lower bound for the equilateral number of normed spaces. *Proc. Amer. Math. Soc.*, 136(1):127–131, 2008. URL: <http://dx.doi.org/10.1090/S0002-9939-07-08916-2>, doi:10.1090/S0002-9939-07-08916-2.

Searching for the closest-pair in a convex polygonal translate

Jie Xue

Department of Computer Science & Engineering, University of Minnesota
Minneapolis, MN, USA
xuexx193@umn.edu
 <http://cs.umn.edu/~xuexx193>

Yuan Li

Facebook Inc.
Seattle, WA, USA
lydxlx@fb.com

Saladi Rahul

Department of Computer Science, University of Illinois
Urbana, IL, USA
saladi.rahul@gmail.com

Ravi Janardan

Department of Computer Science & Engineering, University of Minnesota
Minneapolis, MN, USA
janardan@umn.edu
 <http://cs.umn.edu/~janardan>

1 Introduction

The closest-pair problem is one of the most fundamental problems in computational geometry and finds many applications, e.g., collision detection, similarity search, traffic control, etc. The *range closest-pair* (RCP) problem, as a range-search version of the classical closest-pair problem, aims to preprocess a given set S of points into a data structure such that when a query range $X \in \mathcal{X}$ is specified (where \mathcal{X} is the collection of all possible query ranges, called *query space*), the closest-pair in the subset $S \cap X$ can be reported efficiently. The motivation for the RCP problem is clear and similar to that of range search: in many situations, one is interested in local information (i.e., local closest-pairs) inside specified ranges rather than global information (i.e., global closest-pair) of the dataset.

The RCP problem in \mathbb{R}^2 has been studied in prior work over the last fifteen years, e.g., [1, 2, 3, 4]. The papers [2] and [3] considered the RCP problem for orthogonal queries (specifically, quadrants, strips, rectangles), while [1] mainly studied the RCP problem for halfplane query. Very recently, in the authors' work [4], the RCP problem for these query types was revisited and many previous results were improved. The best known RCP data structure for rectangle queries requires $O(n \log^2 n)$ space and $O(\log^2 n)$ query time [4].

In this paper, we investigate a new variant of the RCP problem in which the query ranges are translates of a convex polygon. Formally, let Γ be a fixed convex polygon in \mathbb{R}^2 called *base range*, and $\mathcal{L}_\Gamma = \{q + \Gamma : q \in \mathbb{R}^2\}$ be the collection of all Γ -translates. We are interested in the RCP problem with query space \mathcal{L}_Γ . Our main result is the following theorem.

► **Theorem 1.** *Let Γ be a fixed convex polygon in \mathbb{R}^2 . Given a dataset S of n points in \mathbb{R}^2 , one can build an $O(n)$ -space data structure on S to answer any RCP query $X \in \mathcal{L}_\Gamma$ in $O(\log n)$ time, where $O(\cdot)$ hides constants depending on Γ .*

We remark that the constants hidden in $O(\cdot)$ in Theorem 1 depends on not only the number of the vertices of Γ but also some other characteristics of Γ , e.g., its angles, the pairwise

distances of the vertices, etc. We have not yet found an efficient algorithm to build the data structure in Theorem 1; we leave this as an open question for future study.

Although we restrict the query ranges to be translates of the base range (which seemingly makes things easier), the problem is still challenging for several reasons. First, the RCP problem is a non-decomposable range-search problem in the sense that even if the query range X can be written as $X = X_1 \cup X_2$, the closest-pair in $S \cap X$ cannot be known from the closest-pairs in $S \cap X_1$ and $S \cap X_2$. The non-decomposability makes many traditional range-search techniques inapplicable to the RCP problem. Second, the base range Γ to be considered is a general convex polygon; we do not make any further assumption about Γ . Finally, we are looking for an *optimal* data structure (i.e., with linear space and logarithmic query time), which is usually difficult for RCP-type problems. Only very recently, two optimal RCP data structures (for quadrant and halfplane queries) were given [4].

2 An overview of our approach

In order to prove Theorem 1, we give a reduction from the RCP problem with Γ -translation queries to the RCP problem with wedge-translation queries, which is our first technical contribution. To this end, we first observe that for a *decomposable* range-search problem (e.g., range reporting), a Γ -translation query can be reduced to $|\Gamma|$ wedge-translation queries by using a grid-based approach (where $|\Gamma|$ is the number of the vertices of Γ). Using this observation, we obtain an optimal range-reporting data structure for Γ -translation queries by considering the range-reporting problem with wedge-translation queries (for which a simple optimal data structure can be easily designed). However, unlike range reporting, the RCP problem is *non-decomposable*. As such, to obtain a similar reduction requires more work. We exploit some properties of the closest-pair in a Γ -translate, and eventually show that to prove Theorem 1, it suffices to design **(i)** an optimal RCP data structure for wedge-translation queries and **(ii)** an optimal range-reporting data structure for Γ -translation queries. We have already obtained **(ii)**, as mentioned above. Therefore, our task is reduced to designing **(i)**, i.e., an optimal RCP data structure for wedge-translation queries.

Let W be a fixed wedge in \mathbb{R}^2 . We need to consider the RCP problem with query space $\mathcal{L}_W = \{q + W : q \in \mathbb{R}^2\}$. However, this is still a nontrivial problem. In order to design an optimal data structure, we prove a combinatorial result claiming that among the $O(n^2)$ point-pairs in S , only $O(n)$ pairs can be the answers of the RCP queries in \mathcal{L}_W (we call them *candidate pairs*), which is our second technical contribution. Our proof first properly separates the candidate pairs into two classes according to how they lie in the minimal W -translates containing them. We then show the linear sizes of the two classes separately by exploiting different geometric insights, which eventually implies that the total number of the candidate pairs is linear. With this in hand, to build the data structure becomes easy. We store all the candidate pairs and use them to construct a planar subdivision D such that finding the closest candidate pair in an RCP query $q + W \in \mathcal{L}_W$ is equivalent to doing a point-location for q in D . It is easy to see that the complexity of D is linear in the number of the candidate pairs, and hence linear in n . Therefore, using an optimal planar point-location data structure, we obtain an optimal RCP data structure for query space \mathcal{L}_W . Combining this with the reduction mentioned above, Theorem 1 is finally proved.

We remark here that some of our techniques might be extended to more general settings in which the base range Γ is a non-convex polygon or a general convex body. We are still in process of investigating the RCP problem with translation queries in these settings.

References

- 1 M. A. Abam, P. Carmi, M. Farshi, and M. Smid. On the power of the semi-separated pair decomposition. In *Workshop on Algorithms and Data Structures*, pages 1–12. Springer, 2009.
- 2 P. Gupta, R. Janardan, Y. Kumar, and M. Smid. Data structures for range-aggregate extent queries. *Computational Geometry: Theory and Applications*, 2(47):329–347, 2014.
- 3 R. Sharathkumar and P. Gupta. Range-aggregate proximity queries. *Technical Report IIT/TR/2007/80. IIT Hyderabad, Telangana*, 500032, 2007.
- 4 J. Xue, Y. Li, S. Rahul, and R. Janardan. New bounds for range closest-pair problems. *Accepted to the 34th International Symposium on Computational Geometry*, 2018.

Weighted Voronoi Diagrams in the L_∞ -Norm

Günther Eder and Martin Held¹

University of Salzburg, Austria

Abstract

We study Voronoi diagrams of n weighted points in the plane in the maximum norm. We establish a tight $\Theta(n^2)$ worst-case combinatorial bound for such a Voronoi diagram and introduce an incremental construction algorithm that allows its computation in $\mathcal{O}(n^2 \log n)$ time.

2012 ACM Subject Classification Theory of computation \rightarrow Proof complexity

Keywords and phrases Weighted Voronoi Diagrams, Maximum Norm, Complexity, Algorithm

1 Introduction and Definition

In 1984 Aurenhammer and Edelsbrunner [1] introduced a worst-case optimal $\mathcal{O}(n^2)$ time algorithm to compute the Voronoi diagram of n multiplicatively weighted point sites in the L_2 metric. We investigate Voronoi diagrams of multiplicatively weighted point sites in the L_∞ metric. Contrary to the L_2 diagram, which consists of circular arcs, the L_∞ diagram is given by a PSGL. There is no obvious way to extend the linear-time half-space intersection of [1], which relies on a spherical inversion, to our setting, i.e., to scaled unit cubes.

Let S denote a finite set of n weighted points, *sites*, in \mathbb{R}^2 and consider a weight function $w: S \rightarrow \mathbb{R}^+$ assigning a weight $w(s)$ to every site. For the sake of descriptiveness we assume all weights of S to be unique. The weighted L_∞ distance $d_w(p, s)$ between an arbitrary point p in \mathbb{R}^2 and a site $s \in S$ is the standard L_∞ distance $d(p, s)$ between p and s divided by the weight of s . For s_i in S , the (*weighted*) *Voronoi region* $\mathcal{R}(s_i)$ is the set of all points of the plane that are closer to s_i than to any other site in S . The multiplicatively weighted Voronoi Diagram $\mathcal{V}^\infty(S)$ is a subdivision of the plane whose faces are given by (the connected components of) the Voronoi regions of all sites of S . The *bisector* of two distinct sites s_i, s_j of S models the set of points that are at the same weighted distance from s_i and s_j . Let $\square_i(t)$ denote the boundary of an axis-aligned square centered at s_i with a side length of $2 \cdot t \cdot w(s_i)$. Let $\mathcal{U}(t)$ be the set of all such n unit squares scaled by t and corresponding weights. Let $\square_i(t), \square_j(t)$ of $\mathcal{U}(t)$ and $w(s_i) < w(s_j)$. At time $t > 0$ these two squares intersect the first time and at time $t' > t$ $\square_j(t')$ contains $\square_i(t)$ for the first time. The bisector of s_i, s_j is traced out along $\square_j(t) \cap \square_j(t')$ between the times t and t' . A degree-two vertex, *joint*, in the bisector occurs whenever at least one vertex of one square crosses a side of another square. Since this can happen at most once for every vertex-side pair, the bisector of two sites forms a star-shaped polygon with a constant number of vertices.

Clearly $\mathcal{V}^\infty(S)$ is formed by portions of bisectors. Thus $\mathcal{V}^\infty(S)$ consists of straight-line segments and forms a PSLG. It contains Voronoi joints as vertices of degree two, and Voronoi nodes as vertices of degree higher than two. Note that our distinct-weight assumption prevents $\mathcal{V}^\infty(S)$ from containing unbounded edges: Let s_i be the site of S with maximum weight. Then there exists a time t_i such that $\square_i(t)$ contains all other squares of $\mathcal{U}(t)$ for all $t > t_i$. Thus, the Voronoi region of s_i is the only unbounded region.

¹ {geder,held}@cosy.sbg.ac.at; Work supported by Austrian Science Fund (FWF) Grant P25816-N15.

2 Combinatorial Complexity of $\mathcal{V}^\infty(S)$ and Algorithm

Aurenhammer and Edelsbrunner [1] show that a multiplicatively weighted Voronoi diagram in the Euclidean metric has $\Theta(n^2)$ faces, edges, and nodes in the worst case. Their example that illustrates the quadratic worst-case lower bound can be adapted easily to our setting, hence establishing a quadratic lower bound for $\mathcal{V}^\infty(S)$ as well. Their proof of the quadratic upper bound proof is tightly connected to their setting and does not apply to $\mathcal{V}^\infty(S)$.

In the following we sketch how we establish a tight upper bound for $\mathcal{V}^\infty(S)$. The basic idea is that we raise $\mathcal{U}(t)$ to \mathbb{R}^3 by assigning a z -coordinate equal to t to every $\square_i(t)$. Then $\mathcal{U}(t)$, for $0 \leq t \leq \infty$, forms n upside-down pyramids whose apices lie on the xy -plane and coincide with their respective site. The slope of such a pyramid depends on the weight: A larger weight corresponds to smaller slope. Let $\widehat{\mathcal{U}}$ denote this pyramid arrangement. We can show that $\mathcal{V}^\infty(S)$ is the minimization diagram of $\widehat{\mathcal{U}}$.

Now let the sites of S be (re-)numbered such that $w(s_i) > w(s_j)$ for $1 \leq i < j \leq n$, and let $S_i := \{s_1, \dots, s_i\}$. Hence, S_i contains all i sites of S with largest weights. We now focus on the combinatorial complexity of $\mathcal{V}^\infty(S)$. Suppose that one constructs the Voronoi region $\mathcal{R}(s_i)$ and merges it with $\mathcal{V}^\infty(S_{i-1})$ to obtain $\mathcal{V}^\infty(S_i)$. Similarly, in $\widehat{\mathcal{U}}$ we can add the respective pyramids incrementally such that $\widehat{\mathcal{U}}_i$ is the arrangement of all pyramids for S_i . We can show that the newly added pyramid P_i for s_i intersects at most a linear number of edges of the lower envelope of $\widehat{\mathcal{U}}_{i-1}$: Since the weight of s_i is smaller than the weights of all sites of S_i , all pyramids of $\widehat{\mathcal{U}}_{i-1}$ have sides with slopes that are smaller than the slope of the four sides of P_i . Now consider the supporting planes of the four sides of P_i . We look at the intersection of $\widehat{\mathcal{U}}_{i-1}$ and one such plane Π . We show that every pyramid of $\widehat{\mathcal{U}}_{i-1}$ forms a totally defined continuous function in this intersection and that any pair of these functions has the same value at most twice. This property helps to establish a linear upper bound on the combinatorial complexity of the lower envelope of $\Pi \cap \widehat{\mathcal{U}}_{i-1}$. Since all four such envelopes imply an overall linear bound we can conclude that inserting the pyramid P_i into $\widehat{\mathcal{U}}_{i-1}$ results in a linear number of edges in $\mathcal{R}(s_i)$, thus establishing the quadratic upper bound for $\mathcal{V}^\infty(S)$.

Next we sketch our incremental construction algorithm. The first site inserted is s_1 and initially $\mathcal{R}(s_1)$ is the xy -plane. In general, $\mathcal{R}(s_i)$ relative to S_i forms a star-shaped polygon with s_i in its kernel: As stated above, the bisector of two sites s_i, s_j , where $w(s_i) < w(s_j)$, forms a star-shaped polygon of constant combinatorial complexity around s_i . Hence, the intersection of these $i-1$ polygons that model the bisectors between s_i and all sites of S_{i-1} is again a star-shaped polygon with s_i in its kernel: It is $\mathcal{R}(s_i)$ relative to S_i . We can compute such a star-shaped polygon in $\mathcal{O}(n \log n)$ time using a simple divide&conquer approach. As established above, each such polygon is of at most linear size. Merging $\mathcal{V}^\infty(S_{i-1})$ with $\mathcal{R}(s_i)$ takes $\mathcal{O}(n \log n)$ time when utilizing a search structure that is at most quadratic in size; it holds the order of segments that lie on a common line. Finally we delete the edges of $\mathcal{V}^\infty(S_{i-1})$ that lie strictly in the interior of $\mathcal{R}(s_i)$. Let k_i be the number of edges of $\mathcal{V}^\infty(S_{i-1})$ strictly inside of $\mathcal{R}(s_i)$. Then $K := \sum_{0 < i \leq n} k_i \subseteq \mathcal{O}(n^2)$. This claim holds as K can be bounded by the number of edges created during the incremental construction, which in turn is bounded by the combinatorial complexity of $\mathcal{V}^\infty(S_i)$ which is in $\Theta(i^2)$.

► **Theorem 2.1.** *An incremental construction allows to compute $\mathcal{V}^\infty(S)$ of a set S of n weighted sites in $\mathcal{O}(n^2 \log n)$ time and $\mathcal{O}(n^2)$ space.*

References

- 1 F. Aurenhammer and H. Edelsbrunner. An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane. *Pattern Recogn.*, 17(2):251 – 257, 1984.

A technique for polygon inclusion problem

Kai Jin

Department of Computer Science, University of Hong Kong, Pokfulam, Hong Kong SAR
cscjkk@gmail.com

 <https://orcid.org/0000-0003-3720-5117>

Zhiyi Huang

Department of Computer Science, University of Hong Kong, Pokfulam, Hong Kong SAR
hzhyyi.tcs@gmail.com

 <https://orcid.org/0000-0003-2963-9556>

Abstract

The widely known linear time algorithm for computing the maximum area triangle in a convex polygon was found incorrect recently by Keikha et. al. [7]. We present an alternative algorithm in this paper. Comparing to the only previously known correct solution [3], ours is much simpler and more efficient. More importantly, our new approach is powerful in solving related problems.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Maximum Triangles, Minimum all-flush triangle, Rotate-and-Kill

Related Version <https://arxiv.org/abs/1707.04071> and <https://arxiv.org/abs/1712.05081>

1 Introduction

Computing the maximum-area triangles in a convex polygon P is a fundamental problem in computational geometry, and is of significance because follow-up results (e.g. algorithms for polygon collision detection) may use it as a preprocessing step. Surprisingly, two widely known linear time algorithms [4, 2] for computing these triangles were found to be incorrect recently by Keikha et. al. [7]. Toward a correct linear solution, [7] introduced the *3-stable triangles*, i.e. those triangles whose corners lie at P 's vertices and which cannot be enlarged by adjusting one corner. There are $O(n)$ such triangles since they are pairwise *interleaving* (see [2, 7]), where n is the number of vertices in P . Yet computing them in $O(n)$ time was left as an open problem in [7] and has attracted several researchers since then. However, Keikha later found and pointed us to a correct linear time algorithm of Chandran and Mount [3].

In this abstract, by using a technique which we call *Rotate-and-Kill*, we present a new linear time algorithm for computing the 3-stable triangles, which has advantages over Chandran and Mount's algorithm in every aspect. According to our implementation of both algorithms (<https://drive.google.com/open?id=1jnaqwhTgN4EB6kXXIXqqX1VN1-9Ct6M1>), our algorithm is simpler (4 times shorter in code length), more efficient (approximately 30 times faster), and much more robust (it easily solves the case $n = 10^7$, whereas the previous algorithm fails even if $n = 1000$ due to accuracy problems of floats). More interestingly, our technique extends to solve other polygon inclusion and circumscribing problems. For example, we adapt it to compute in linear time (i) all the *general 3-stable triangles* (whose definition is similar as that of 3-stable triangles, except changing the requirement "lie at P 's vertices" to "lie at P 's boundary"), (ii) all the minimum-area triangles enclosing P , and (iii) all the Minimum-area all-Flush Triangles (MFTs) (each side in which must contain an edge of P) enclosing P . Note that the last algorithm for computing the MFTs improves over the state-of-art algorithms [1, 8] by a factor of $\log n$. This algorithm is presented in [6], whereas the introduction of the Rotate-and-Kill and the other algorithms are presented in [5].

Notation. Let v_1, \dots, v_n be a clockwise enumeration of the vertices of P . When A denotes a vertex of P , let $A-1, A+1$ respectively denote the clockwise previous and next vertex of A . Let \overleftrightarrow{AB} denote the line connecting A, B . We start our algorithm by computing one 3-stable triangle $\triangle v_r v_s v_t$ (details omitted due to space limit; see Section 2 in [5]), where v_r, v_s, v_t lie in clockwise order. Note that each 3-stable triangle has one corner belonging to $\{v_s, \dots, v_t\}$ and has the clockwise next corner belonging to $\{v_t, \dots, v_r\}$ (because it interleaves $\triangle v_r v_s v_t$).

Rotate-and-Kill process for computing all 3-stable triangles. Initially, set $(B, C) = (v_s, v_t)$. In each iteration, we first compute A so that A has the largest distance to \overleftrightarrow{BC} among vertices on the right of \overleftrightarrow{BC} . This only costs amortized $O(1)$ time because the slope of BC will keep decreasing and so A goes only in clockwise direction. Then, check whether $\triangle ABC$ is 3-stable and report it if so. Then, proceed to the next iteration by either killing B (i.e. moving pointer B to its next vertex) or killing C (i.e. moving pointer C to its next vertex). Whether B or C should be killed is decided according to a particular **criterion**. The important thing is that B is killed only when its related pairs $(B, C+1), (B, C+2), \dots, (B, v_r)$ cannot form an edge of any 3-stable triangle; and C is killed only when its related pairs $(B+1, C), (B+2, C), \dots, (v_t, C)$ cannot form an edge of any 3-stable triangle. This implies that our algorithm will not miss any 3-stable triangles. We note that such a criterion must exist according to the fact that all 3-stable triangles are interleaving; see the proof of Observation 5 in [5]; but it would be useless if it is not computational efficient. In fact, to design a linear time algorithm, the core of our approach lies in designing a good criterion that can be computed in constant time. Interestingly, as presented below, such a good criterion also exists! Finally, the Rotate-and-Kill process is terminated when (B, C) reaches (v_t, v_r) .

```

1  $(A, B, C) \leftarrow (v_r, v_s, v_t)$ ;
2 repeat
3   | Let  $A$  be the vertex furthest to  $\overleftrightarrow{BC}$  on the right of  $\overleftrightarrow{BC}$ . Output  $ABC$  if it is 3-stable;
4   | if  $A > I_{B,C}$  in the distance to  $\overleftrightarrow{BC}$  then  $C \leftarrow C + 1$ ; else  $B \leftarrow B + 1$ ;
5 until  $(B, C) = (v_t, v_r)$ ;
```

The criterion is simply “ $A > I_{B,C}$ ”, where $I_{B,C}$ is some point that is easy to compute.

Summary. We present a powerful technique for solving both inclusion and circumscribing extremal polygon problems. This technique is distinct from the well-known Rotating-caliper technique [9]. Instead, it can be regarded as a corrective version of the wrong technique of Dobkin and Snyder [4]. Nonsurprisingly, some extra efforts might be necessary in applying our technique to any specific problem; a typical application is demonstrated in [6] as stated.

The methods of this abstract suggest that more efficient algorithms may exist for some related problems. In future work, we want to attempt applying our technique to compute the following extremal polygons: (a) the maximum-area 4-gons (currently best algorithm works in $O(n \log n)$ time); (b) the maximum-perimeter triangles (currently best algorithm works in $O(n \log n)$ time); (c) the maximum equilateral triangle (currently best algorithm works in $O(n^2)$ time). Note that the first two types of extremal polygons also admit the pairwise-interleaving property, which is a prerequisite of applying our technique.

References

- 1 A. Aggarwal, B. Schieber, and T. Tokuyama. Finding a minimum-weight k -link path in graphs with the concave monge property and applications. *Discrete & Computational Geometry*, 12(1):263–280, 1994.

- 2 J. E. Boyce, D. P. Dobkin, R. L. (Scot) Drysdale, III, and L. J. Guibas. Finding extremal polygons. In *14th Symposium on Theory of Computing*, pages 282–289, 1982.
- 3 S. Chandran and D. M. Mount. A parallel algorithm for enclosed and enclosing triangles. *International Journal of Computational Geometry & Applications*, 02(02):191–214, 1992. doi:10.1142/S0218195992000123.
- 4 D. P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 9–17, Oct 1979. doi:10.1109/SFCS.1979.28.
- 5 Kai Jin. Maximal area triangles in a convex polygon. *CoRR*, abs/1707.04071, 2017.
- 6 Kai Jin and Zhiyi Huang. Minimum area all-flush triangles circumscribing a convex polygon. *CoRR*, abs/1712.05081, 2017.
- 7 V. Keikha, M. Löffler, J. Urhausen, and I. v. d. Hoog. Maximum-area triangle in a convex polygon, revisited. *CoRR*, abs/1705.11035, 2017.
- 8 B. Schieber. Computing a minimum-weight k-link path in graphs with the concave monge property. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 405–411. Society for Industrial and Applied Mathematics, 1995.
- 9 G. Toussaint. Solving geometric problems with the rotating calipers. In *In Proc. IEEE MELECON'83*, pages 10–02, 1983.

Delone Sets for Convex Bodies

Ahmed Abdelkader

Department of Computer Science
University of Maryland, College Park MD, USA
akader@cs.umd.edu

 <https://orcid.org/0000-0002-6749-1807>

Abstract

We demonstrate an intriguing application of Delone sets to the approximation of convex bodies. With the help of Macbeath regions, it is possible to construct an ε -approximation of any convex body in \mathbb{R}^d as the union of $O(1/\varepsilon^{(d-1)/2})$ ellipsoids, where the center points of these ellipsoids form a Delone set in the Hilbert metric defined intrinsically on the convex body. A hierarchy of such approximations yields a data structure that answers ε -approximate polytope membership queries in $O(\log 1/\varepsilon)$ time, matching the best asymptotic results for this problem by a data structure that is both simpler and arguably more elegant.

2012 ACM Subject Classification Theory of computation \rightarrow Randomness, geometry and discrete structures \rightarrow Computational geometry

Keywords and phrases sampling, convex, approximation, membership, query

Acknowledgements The author thanks David Mount for numerous useful discussions.

1 Introduction

Let K denote a full-dimensional convex polytope presented as the intersection of n halfspaces in \mathbb{R}^d , where d is a fixed constant. The objective is to preprocess K so that, given any query point $q \in \mathbb{R}^d$, it is possible to determine efficiently whether q lies within K . In particular, for $\varepsilon > 0$, an ε -approximate polytope membership (APM) query returns a positive result if $q \in K$, a negative result if the distance from q to its closest point in K is greater than $\varepsilon \cdot \text{diam}(K)$, and it may return either result otherwise. Polytope membership queries arise in many application areas, such as linear programming, ray-shooting queries, nearest neighbor searching, collision detection, and machine learning.

Following a series of results, a space-optimal solution in the polylogarithmic query time regime was presented in [1]. By abandoning grids and quadtrees in favor of a new paradigm based on ellipsoids derived from *Macbeath regions (M-regions)* [4], APM queries can be answered in $O_d(\log \frac{1}{\varepsilon})$ time using $O_d(1/\varepsilon^{(d-1)/2})$ storage (O_d hides factors exponential in d).

Given a point $x \in K$, and $\lambda \geq 0$, the λ -scaled *M-region* at x , denoted $M_K^\lambda(x)$, is defined as $x + \lambda((K - x) \cap (x - K))$. Namely, $M_K^\lambda(x)$ is the intersection of K with its reflection about x . It follows that $M_K^\lambda(x)$ is convex and centrally symmetric about x . Hence, it is often more convenient to replace $M_K^\lambda(x)$ with a suitable ellipsoid centered at x .

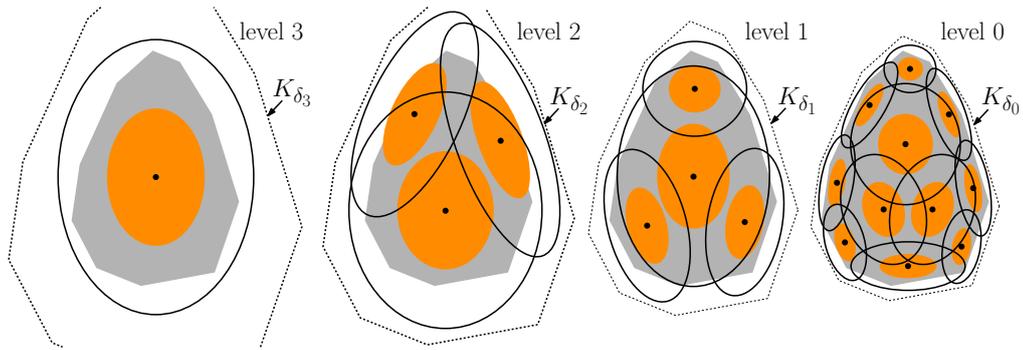
The approach presented in [1] is based on constructing layers of nested eroded bodies within K and covering the boundaries of these eroded bodies with ellipsoids that are based on M-regions. An APM query is answered by shooting a ray from a central point in the polytope towards the boundary of K , and tracking an ellipsoid at each layer that is intersected by the ray. While it is asymptotically optimal [1], the data structure and its analysis are complicated by peculiarities of ray-shooting. In this abstract, we propose a simpler and more intuitive approach based on *Delone sets*.

2 Overview

A key property of M-regions is that if two shrunken M-regions overlap, then an appropriate expansion of one contains the other [3]. This *expansion-containment* property bears similarity to packing-covering properties of *Delone sets*. The connection is elucidated by observing that M-regions are closely related to the metric balls B_H of the *Hilbert metric* defined on the convex body [2]. The following is a generalization of the result in [2].

► **Lemma 1.** $\forall x \in K$ and $0 \leq \lambda < 1$, $B_H(x, \frac{1}{2} \ln(1 + \lambda)) \subseteq M_K^\lambda(x) \subseteq B_H(x, \frac{1}{2} \ln \frac{1+\lambda}{1-\lambda})$.

Hence, M-regions may serve as proxies of intrinsic metric balls. Our constructions involve expanding K , but the nature of expansion K_δ is flexible. Recall that K is in κ -canonical form if $B(\frac{\kappa}{2}) \subseteq K \subseteq B(\frac{1}{2})$, with B a ball centered at $\mathbf{0}$. The constant packing-covering scales used to define the Delone set will be denoted by $0 < \lambda_p < \lambda_c < 1$. Consider maximal sets of disjoint M-regions $M_\delta^{\lambda_p}(x)$ defined with respect to K_δ , such that the centers x lie within K . Let X_δ denote such a set of centers. When refining to smaller expansions, a bound on the growth rate in the number of M-regions is crucial. Let $Y_{\delta,s}(x)$ denote the set of points y such that $M_\delta^{\lambda_p}(y)$ are pairwise disjoint and overlap $M_{s\delta}^{\lambda_c}(x)$ for $\delta \geq 0$ and $s \geq 1$. By layering the Delone sets X_{δ_i} , with $\delta_i = 2^i \varepsilon$, we obtain a DAG structure.



■ **Figure 1** Hierarchy of Delone sets for APM: packing λ_p (orange) and coverage λ_c (transparent).

► **Lemma 2.** If K is in κ -canonical form, $\delta \geq 0$ and $\lambda_p \leq 1/5$, then $|X_\delta| = O_d(1/\delta^{(d-1)/2})$.

► **Lemma 3.** $\forall x \in K$, $\delta \geq 0$ and $s \geq 1$, $|Y_{\delta,s}(x)| = O_d(1)$.

Given a query point $q \in \mathbb{R}^d$, we start at the root node of the DAG and descend to any child node associated with some $x \in K$ such that $q \in M_\delta^{\lambda_c}(x)$ or abort if none exists. If we reach an M-region $M_\varepsilon^{\lambda_c}(x) \ni q$, then $q \in K_\varepsilon$. With $O(\log 1/\varepsilon)$ height and $O_d(1)$ branching factor, the query takes $O_d(\log 1/\varepsilon)$ time while storage is dominated by $|X_\varepsilon|$.

References

- 1 S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal approximate polytope membership. In *Proc. 28th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 2017.
- 2 C. Walsh C. Vernicos. Flag-approximability of convex bodies and volume growth of Hilbert geometries. HAL Archive (hal-01423693i), 2016.
- 3 G. Ewald, D. Larman, and C. Rogers. The directions of the line segments and of the r -dimensional balls on the boundary of a convex body in Euclidean space. *Mathematika*, 17, 1970.
- 4 A. M. Macbeath. A theorem on non-homogeneous lattices. *Ann. of Math.*, 56, 1952.

$\mathcal{O}(k)$ -robust spanners in one dimension

Kevin Buchin

Eindhoven University of Technology, Eindhoven, The Netherlands
k.a.buchin@tue.nl

Tim Hulshof

Eindhoven University of Technology, Eindhoven, The Netherlands
w.j.t.hulshof@tue.nl

Dániel Oláh

Eindhoven University of Technology, Eindhoven, The Netherlands
d.olah@tue.nl

1 Introduction

Geometric networks are graphs whose vertices are points in the d -dimensional Euclidean space \mathbb{R}^d ($d \geq 1$) and whose edges are weighted by the Euclidean distance between its endpoints. We say that a geometric network $G = (V, E)$ is a t -spanner of $V' \subseteq V$ for some $t \geq 1$ if for all $x, y \in V'$, $d_G(x, y) \leq t \cdot d(x, y)$ holds, where $d_G(x, y)$ is the length of the shortest path between x and y and $d(x, y)$ is the Euclidean distance between x and y . Spanners have been studied extensively and there are many constructions for spanners with various properties [3].

One such property is the resistance of t against vertex failures. To this end, Bose et al. [1] introduced the following notion of robustness. Let $G = (V, E)$ be a (geometric) t -spanner for some $t \geq 1$ and let $f : \mathbb{N} \rightarrow \mathbb{N}$ be an arbitrary function. Then we say that G is an $f(k)$ -robust t -spanner if for any subset F of V ($k = |F|$), there exists a set $F^* \supseteq F$ with $|F^*| \leq f(k)$, such that the subgraph induced by $V \setminus F$ is a t -spanner of $V \setminus F^*$. Intuitively, a spanner is robust if the removal of a few vertices can only harm a small number of other vertices.

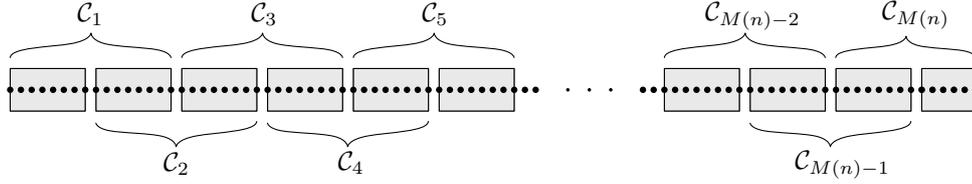
An open problem posed by Bose et al. [1] is the construction of $\mathcal{O}(k)$ -robust spanners. They prove that a lower bound for the size of $\mathcal{O}(k)$ -robust spanners on n vertices is $\Omega(n \log n)$ edges, but give no upper bound below the trivial $\mathcal{O}(n^2)$ edges.

In this note we present a construction for an $\mathcal{O}(k)$ -robust 1-spanner on n vertices of size $\mathcal{O}(n^{\frac{3}{2}})$ for any one-dimensional point set. We also show a generalization that improves this upper bound to $\mathcal{O}(n^{1+\varepsilon})$ edges for any one-dimensional point set and any $\varepsilon > 0$.

2 A simple construction

Let $V = \{x_1, x_2, \dots, x_n\}$ be an arbitrary point set with $x_i \in \mathbb{R}$ for $1 \leq i \leq n$ and $x_i < x_{i+1}$ for $1 \leq i \leq n-1$. Let $m \in \mathbb{N}$ be such that $(2m)^2 \leq n < (2m+2)^2$. We construct a graph $G = (V, E)$ as follows. Let $\mathcal{C}_i = \{x_{(i-1)m+1}, x_{(i-1)m+2}, \dots, x_{(i+1)m}\}$ for $1 \leq i \leq 4m-1 + \lceil \frac{n-(2m)^2}{m} \rceil =: M(n)$. We call \mathcal{C}_i the i^{th} cluster. There are $\mathcal{O}(m)$ clusters and each of them has exactly $2m$ vertices, except the last one, which may contain fewer. Note that adjacent clusters are half-overlapping, that is $|\mathcal{C}_i \cap \mathcal{C}_{i+1}| = m$. We define half-clusters that are simply made by splitting each cluster in the middle, see Figure 1. Therefore, the number of half-clusters is $\mathcal{O}(m)$ and each of them (possibly except the last one) contains exactly m vertices. Let \mathcal{H}_i^L and \mathcal{H}_i^R denote the left and right half of \mathcal{C}_i , respectively. Note that $\mathcal{H}_i^L = \mathcal{H}_{i-1}^R$.

We add two types of edges to the graph. First, we form cliques in each cluster \mathcal{C}_i , adding $\mathcal{O}(m^3)$ edges. Second, for any pair of half-clusters we add an arbitrary complete matching between the two half-clusters. The number of edges we add is again $\mathcal{O}(m^3)$. If the size of the last half-cluster is smaller than m , then we do not add matchings for this half-cluster.



■ **Figure 1** The structure of the clusters of G . Grey rectangles indicate half-clusters. Each half-cluster contain m vertices, except the last one that may contain fewer vertices, depending on the actual value of n .

► **Theorem 1.** *For any 1-dimensional point set V (with $|V| = n$), the graph G constructed above is an $\mathcal{O}(k)$ -robust 1-spanner with $\mathcal{O}(n^{\frac{3}{2}})$ edges.*

Proof. Clearly, the size of G is $\mathcal{O}(m^3) = \mathcal{O}(n^{\frac{3}{2}})$. To show that G is an $\mathcal{O}(k)$ -robust 1-spanner, first, it remains to construct the set F^* for any set of failed points F such that $|F^*| = \mathcal{O}(|F|)$. To start, set $F^* = F$. Then for each half-cluster \mathcal{H}_i^L ($2 \leq i \leq M(n)$), if $|\mathcal{H}_i^L \cap F| \geq \frac{m}{2}$, we update F^* by adding the clusters \mathcal{C}_{i-1} and \mathcal{C}_i . Thus $|F^*| \leq 6 \cdot |F|$.

Secondly, we have to prove that the subgraph induced by $V \setminus F$ is a 1-spanner of $V \setminus F^*$. Let $x, y \in V \setminus F^*$ (with $x < y$). There are three cases. First, if x and y are in the same cluster, then $\{x, y\} \in E$ and the claim holds. Second, if they are not in the same cluster, but in overlapping clusters, then there is a vertex $z \notin F$ in the overlap that shares an edge with both x and y . The third case is when $x \in \mathcal{H}_i^L$ and $y \in \mathcal{H}_j^R$, with $i < j + 1$. Then, we know that $|\mathcal{H}_i^R \cap F| < \frac{m}{2}$, otherwise $x \in F^*$. Similarly, we know that $|\mathcal{H}_j^L \cap F| < \frac{m}{2}$, otherwise $y \in F^*$. Therefore, by the pigeonhole principle there is a vertex $x' \in \mathcal{H}_i^R \setminus F$ and a vertex $y' \in \mathcal{H}_j^L \setminus F$ for which there is an edge $\{x', y'\} \in E$ from the matching between \mathcal{H}_i^R and \mathcal{H}_j^L . It is clear that $\{x, x'\} \in E$ and $\{y, y'\} \in E$, since within a cluster all vertices are connected. The length of this path between x and y is $d(x, y)$, because $x < x' < y' < y$ by construction. ◀

3 Iterated construction

We generalize the construction by introducing an additional parameter ℓ that determines the number of layers in the construction of $G_\ell = (V, E)$. The case $\ell = 1$ corresponds to the construction above. In each layer all clusters have the same size, namely, the clusters in layer i have size $n^{\frac{i}{\ell+1}}$ for $1 \leq i \leq \ell$ and they are half-overlapping as before. Let $\mathcal{C}_{i,j}$ denote the j^{th} cluster in layer i . For any cluster $\mathcal{C}_{1,j}$ in the first (lowest) layer, form a clique on its vertices. In layer i ($2 \leq i \leq \ell$), for any cluster $\mathcal{C}_{i,j}$, form a complete matching between any pair of half-clusters from layer $i - 1$ that are contained in $\mathcal{C}_{i,j}$. Finally, add a complete matching between any pair of half-clusters in the top layer. Thus, the size of G is $\mathcal{O}(\ell \cdot n^{\frac{\ell+2}{\ell+1}})$ edges.

► **Theorem 2.** *For any $\varepsilon > 0$ and any 1-dimensional point set V (with $|V| = n$), the graph G_ℓ constructed above is an $\mathcal{O}(k)$ -robust 1-spanner with $\mathcal{O}(n^{1+\varepsilon})$ edges for $\ell \geq \frac{1-\varepsilon}{\varepsilon}$. [2]*

In personal communication Sariel Har-Peled showed us that the sharp upper bound for the size of $\mathcal{O}(k)$ -robust spanners is $\mathcal{O}(n \log n)$.

Acknowledgements. The work in this paper is supported by the Netherlands Organisation for Scientific Research (NWO) through Gravitation-grant NETWORKS-024.002.003.

References

- 1 Prosenjit Bose, Vida Dujmović, Pat Morin, and Michiel Smid. Robust geometric spanners. *SIAM Journal on Computing*, 42(4):1720–1736, 2013.
- 2 Kevin Buchin, Tim Hulshof, and Dániel Oláh. $\mathcal{O}(k)$ -robust spanners in one dimension, 2018. [arXiv:1803.08719](https://arxiv.org/abs/1803.08719).
- 3 Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, New York, NY, USA, 2007.

Peeling Digital Potatoes

Loïc Crombez

Université Clermont Auvergne and LIMOS, Clermont-Ferrand, France

Guilherme D. da Fonseca

Université Clermont Auvergne and LIMOS, Clermont-Ferrand, France

Yan Gérard

Université Clermont Auvergne and LIMOS, Clermont-Ferrand, France

1 Introduction

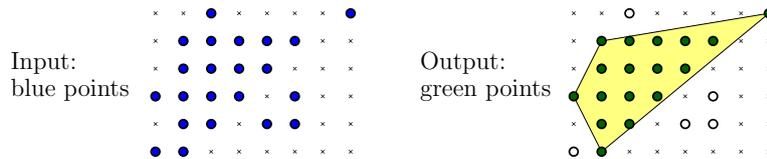
The *potato-peeling problem* (also called the *convex skull problem*) consists of finding the largest convex polygon enclosed in a given input polygon. Unlike the related convex hull problem that can be solved in $O(n \log n)$ time, the fastest known algorithm solves the problem in $O(n^7)$ time [2]. Herein, we consider a digital variation of the potato-peeling problem.

Digital geometry is the field of mathematics that studies the geometry of points with integer coordinates, also known as *lattice points* [5]. A set of lattice points $K \subset \mathbb{Z}^2$ is *digital convex* if $\text{conv}(K) \cap \mathbb{Z}^2 = K$, where $\text{conv}(K)$ denotes the convex hull of K . The *digital potato-peeling problem* is defined as follows.

DIGITAL POTATO-PEELING PROBLEM

Input: Set $S \subset \mathbb{Z}^2$ of n lattice points given by their coordinates.

Question: Determine the *largest* set $K \subseteq S$ that is digital convex, i.e. $\text{conv}(K) \cap \mathbb{Z}^2 = K$.



In this paper, we present an exact algorithm that solves the problem in $O(n^3 \log nr)$ time, where r is the diameter of S . Our algorithm works for different versions of the problem, in which we define the largest convex polygon in terms of area, cardinality, or perimeter. More generally, the algorithm works for any measure f such that for any two convex polygons $P \subseteq P'$, we have $f(P) \leq f(P')$. Heuristics for the digital potato-peeling problem have been presented in [1, 3].

We remark that an algorithm for the digital potato-peeling problem does not yield an algorithm for the continuous problem or vice versa. In the digital version of the problem, only the n input points may be vertices of the solution, which is not true in the continuous case. On the other side, the definition of digital convex is more subtle than standard convexity and is connected to the geometry of numbers.

2 Algorithm

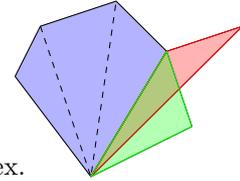
A digital convex set can be described by its convex hull, furthermore the vertices of this hull are lattice points. Hence, instead of explicitly looking for K , our algorithm looks for $\text{conv}(K)$. It is always possible to triangulate a convex polygon with k vertices using $k - 2$ triangles that share the bottom-most vertex p . Our algorithm uses the fact that such triangulation exists. For each point $p \in S$, the algorithm determines the largest convex polygon containing p as a bottom-most vertex. The algorithm proceeds by appending the aforementioned triangles from left to right using dynamic programming.

We say that a triangle T is *valid* if $T \cap \mathbb{Z}^2 = T \cap S$. The algorithm uses a data structure for triangle range counting for S with $O(n^{2+\varepsilon})$ storage and preprocessing time and $O(\log n)$ query time [4]. The number of lattice points in a triangle T can be determined with Pick's theorem and Euclid's algorithm in $O(\log r)$ time, where r is the diameter of T . We can therefore determine if a triangle T is valid in $O(\log nr)$ time by comparing the two results. Only the valid triangles are considered by the algorithm.

Next, we fix p and consider the problem of finding the largest digital convex polygon K that has p as the bottom-most vertex. Without loss of generality, we assume that p is the lowest point in S . Given three points p_1, p, p_2 , let $\angle p_1, p, p_2$ denote the positive clockwise angle between p_1 and p_2 around p . We sort the points $q \in S$ by $\angle(p - (1, 0)), p, q$. This sorts the points in clockwise order around p . Let p_1, \dots, p_n denote the points in sorted order.

Let $\Delta_{i,j}$ denote the triangle whose vertices are p, p_i, p_j . For all $p_i, p_j \in S$ with $i < j$ and such that $\Delta_{i,j}$ is valid, the algorithm determines the largest convex polygon that has $\Delta_{i,j}$ as the right-most triangle. We refer to this convex polygon as $C_{i,j}$. The key property to efficiently compute $C_{i,j}$ is

$$C_{i,j} = \Delta_{i,j} \cup \max_h C_{h,i}, \text{ where } h < i \text{ is such that } \angle p_j, p_i, p_h \text{ is convex.}$$



A naïve dynamic programming algorithm to compute $C_{i,j}$ takes $O(n^3 + n^2 \log nr)$ time, since there are $O(n^2)$ values of $C_{i,j}$ to be calculated and each calculation involves computing the maximum among $O(n)$ previously calculated values. The total running time to apply this method to all points $p \in S$ is therefore $O(n^4 + n^3 \log nr)$. Next, we show how to reduce the running time to $O(n^3 \log nr)$.

Instead of looking for the largest possible convex polygon for each triangle, the principle is, for a given i , to consider every $C_{h,i}$ in decreasing order according to their size, and append them to every valid $\Delta_{i,j}$ that satisfies the angle constraint. To find those triangles efficiently, we sort them according to the orientation of their $p_i p_j$ edge. For each i , we start by sorting $\mathcal{C} = \{C_{h,i} : h < i\}$ by decreasing size and $\mathcal{T} = \{\Delta_{i,j} : i < j\}$ by decreasing $\angle(p_i - (1, 0)), p_i, p_j$. Then, for all left convex $C_{h,i}$ in \mathcal{C} , in decreasing order, we append all the right triangles $\Delta_{i,j}$ from \mathcal{T} that preserve convexity. Those triangles are contiguous and at the beginning of \mathcal{T} thanks to the sorting.

Sorting the two lists $\{C_{h,i} : h < i\}$ and $\{\Delta_{i,j} : i < j\}$ takes at most $O(n \log n)$ time for each point p_i and testing if a triangle is valid takes $O(\log nr)$ time. For all p and p_i , the total running time is therefore $O(n^3 \log nr)$.

References

- 1 Gunilla Borgefors and Robin Strand. An approximation of the maximal inscribed convex set of a digital object. In *13th International Conference on Image Analysis and Processing - ICIAP 2005*, pages 438–445, 2005.
- 2 J. S. Chang and C. K. Yap. A polynomial solution for the potato-peeling problem. *Discrete & Computational Geometry*, 1(2):155–182, 1986.
- 3 Jean-Marc Chassery and David Coeurjolly. Optimal shape and inclusion: open problems. In *Mathematical Morphology: 40 Years On, International Symposium on Mathematical Morphology*, Computational Imaging and Vision. Springer Verlag, 2005.
- 4 Bernard Chazelle, Micha Sharir, and Emo Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8(1-6):407–429, 1992.
- 5 Reinhard Klette and Azriel Rosenfeld. *Digital geometry: Geometric methods for digital picture analysis*. Elsevier, 2004.

Constant Approximation Algorithms for Guarding Simple Polygons using Edge and Perimeter Guards

Pritam Bhattacharya

Dept of Computer Science & Engineering, Indian Institute of Technology, Kharagpur, India

Subir K. Ghosh

Dept of Computer Science, RKM Vivekananda Educational & Research Institute, Belur, India

Sudebkumar P. Pal

Dept of Computer Science & Engineering, Indian Institute of Technology, Kharagpur, India

The art gallery problem enquires about the least number of guards sufficient to ensure that an art gallery, represented by a polygon P , is fully guarded, assuming that a guard's field of view covers 360° as well as unbounded distance. An art gallery can be viewed as an n -sided polygon P , with or without holes, and guards as points inside P . Any point $z \in P$ is said to be *visible* from a guard g if the line segment zg does not intersect the exterior of P . Stationary guards that may be placed anywhere inside P are referred to as *point guards*. If stationary guards are allowed to be placed only on the vertices or on the boundary of P , then they are called *vertex guards* and *perimeter guards* respectively. If mobile guards are allowed to patrol only along the edges of P , they are called *edge guards* [7].

Most variants of the art gallery problem have been shown to be NP-hard [2, 10, 11], and recently ETR-complete [1]. In 1987, Ghosh [6, 8] provided a deterministic $\mathcal{O}(\log n)$ -approximation algorithm for vertex and edge guards by discretizing the input polygon P and treating it as an instance of the Set Cover problem. Eidenbenz et al. [5] proved that, if P is allowed to contain holes, then there cannot exist a polynomial time algorithm for the problem with an approximation ratio better than $((1 - \epsilon)/12) \ln n$ for any $\epsilon > 0$, unless $\text{NP} \subseteq \text{TIME}(n^{\mathcal{O}(\log \log n)})$. This inapproximability result establishes that the approximation ratio of $\mathcal{O}(\log n)$ obtained by Ghosh is in fact the best possible for the case of polygons with holes. However, for simple polygons without holes, the existence of a constant factor approximation algorithm for vertex and edge guards was conjectured by Ghosh [8] in 1987. Ghosh's conjecture has been shown to be true for vertex guarding in two special sub-classes of simple polygons, viz. monotone polygons and polygons weakly visible from an edge. Krohn and Nilsson [9] presented an approximation algorithm that computes in polynomial time a guard set for a monotone polygon P , such that the size of the guard set is at most $30 \times \text{OPT}$. Recently, Bhattacharya, Ghosh and Roy [3, 4] presented a 6-approximation algorithm that runs in $\mathcal{O}(n^2)$ time for guarding polygons weakly visible from an edge using vertex guards.

In a very recent paper [?], we obtained three polynomial-time algorithms with a constant approximation ratio for guarding an n -sided simple polygon P using vertex guards. The first algorithm, that has an approximation ratio of 18, guards all vertices of P in $\mathcal{O}(n^4)$ time. The second algorithm, that has the same approximation ratio of 18, guards the entire boundary of P in $\mathcal{O}(n^5)$ time. The third algorithm, that has an approximation ratio of 27, guards all interior and boundary points of P in $\mathcal{O}(n^5)$ time. The significance of our results lies in the fact that these results settle the *long-standing conjecture by Ghosh* [6] regarding the existence of constant-factor approximation algorithms for this problem, which has been open since 1987 despite several attempts by researchers. Following techniques similar to those used to derive the results mentioned above, we also obtained the following results for computing edge guard covers for an n -sided polygon P .

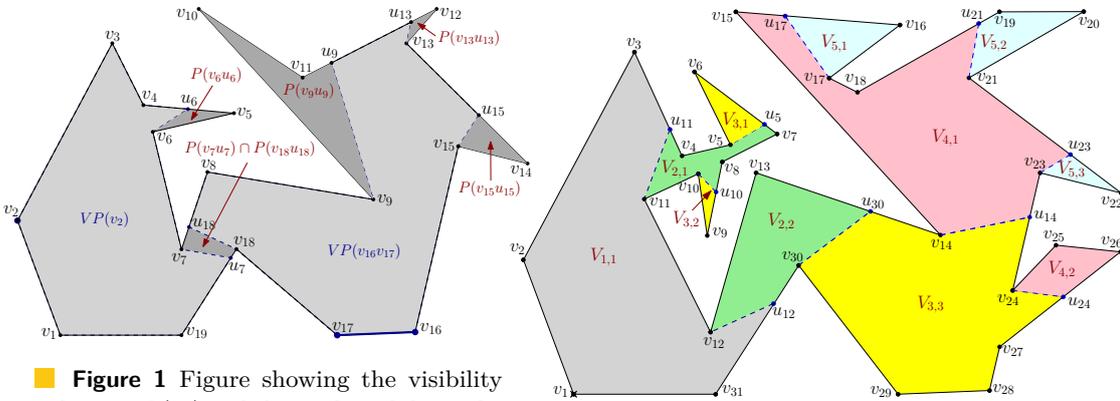
► **Theorem 1.** *A set E of edge guards for guarding all vertices of P can be computed in*

$\mathcal{O}(n^4)$ time, such that $|E| \leq 18 \times |G_{opt}|$, where G_{opt} is a an optimal edge guard cover for all vertices of P .

► **Theorem 2.** A set E of edge guards for guarding the entire boundary of P can be computed in $\mathcal{O}(n^5)$ time, such that $|E| \leq 18 \times |G_{opt}|$, where G_{opt} is a an optimal edge guard cover for the entire boundary of P .

► **Theorem 3.** A set E of edge guards E for guarding the entire interior and boundary of P can be computed in $\mathcal{O}(n^5)$ time, such that $|E| \leq 27 \times |G_{opt}|$, where G_{opt} is a an optimal edge guard cover for the entire interior and boundary of P .

Further, we consider the placement of perimeter guards, which are stationary guards on the boundary of P , but not necessarily at the vertices. We conjecture that, by constructing a modified polygon P' that has $\mathcal{O}(n^2)$ vertices, and then executing our algorithm for finding an edge guard cover on this modified polygon P' , a set S of perimeter guards for guarding all vertices of P can be computed in $\mathcal{O}(n^8)$ time, such that $|S| \leq 18 \times |G_{opt}|$, where G_{opt} is an optimal perimeter guard cover for all vertices of P .



■ **Figure 1** Figure showing the visibility polygon $\mathcal{VP}(v_2)$ and the weak visibility polygon $\mathcal{VP}(v_{16}v_{17})$, with several pockets created by constructed edges belonging to both.

■ **Figure 2** Figure showing the partitioning of a simple polygon into visibility windows.

In each of our algorithms, P is first partitioned into a hierarchy of *weak visibility polygons* according to *link distances* from a starting vertex (see Figure 2), which is very similar to the *window partitioning* given by Suri [12, 13] in the context of computing minimum link paths. Then, starting with the farthest level in the hierarchy (i.e. the set of weak visibility polygons having the maximum link distance from the starting vertex), the entire hierarchy is traversed backward level by level, and at each level, vertex guards (of two types, viz. *inside* and *outside*) are placed for guarding every weak visibility polygon at that level of P . At every level, a novel procedure is used that has been developed for placing guards in (i) a simple polygon that is weakly visible from an internal chord, or (ii) a union of overlapping polygons that are weakly visible from multiple disjoint internal chords. Note that these chords are actually the constructed edges introduced during the hierarchical partitioning of P . Due to partitioning according to link distances, guards can only see points within the adjacent weak visibility polygons in the hierarchical partitioning of P . This property locally restricts the visibility of the chosen guards, and thereby ensures that the approximation bound on the number of vertex guards placed by our algorithms at any level leads directly to overall approximation bounds for guarding P . Thus, a constant factor approximation bound on the overall number of guards placed by our algorithms is a direct consequence of choosing vertex guards in a judicious manner for guarding each collection of overlapping weak visibility polygons obtained from the partitioning of P . Our algorithms exploit several deep visibility structures of simple polygons which are interesting in their own right.

References

- 1 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is $\exists\mathbb{R}$ -complete. *CoRR*, abs/1704.06969, 2017. URL: <http://arxiv.org/abs/1704.06969>, arXiv:1704.06969.
- 2 Alok Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects*. PhD thesis, The Johns Hopkins University, Baltimore, Maryland, 1984.
- 3 Pritam Bhattacharya, Subir Kumar Ghosh, and Bodhayan Roy. Vertex Guarding in Weak Visibility Polygons. In *Proceedings of the 1st International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2015)*, volume 8959 of *Lecture Notes in Computer Science (LNCS)*, pages 45–57. Springer, 2015. URL: http://dx.doi.org/10.1007/978-3-319-14974-5_5, doi:10.1007/978-3-319-14974-5_5.
- 4 Pritam Bhattacharya, Subir Kumar Ghosh, and Bodhayan Roy. Approximability of guarding weak visibility polygons. *Discrete Applied Mathematics*, 228:109 – 129, 2017. URL: <http://www.sciencedirect.com/science/article/pii/S0166218X16306333>, doi:<https://doi.org/10.1016/j.dam.2016.12.015>.
- 5 Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001. URL: <http://link.springer.com/article/10.1007/s00453-001-0040-8>.
- 6 Subir Kumar Ghosh. Approximation algorithms for art gallery problems. In *Proceedings of Canadian Information Processing Society Congress*, page 429–434. Canadian Information Processing Society, 1987.
- 7 Subir Kumar Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007. URL: <http://ebooks.cambridge.org/ref/id/CB09780511543340>, doi:10.1017/CB09780511543340.
- 8 Subir Kumar Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010.
- 9 Erik A Krohn and Bengt J Nilsson. Approximate Guarding of Monotone and Rectilinear Polygons. *Algorithmica*, 66:564–594, 2013. doi:10.1007/s00453-012-9653-3.
- 10 D.T. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1057165.
- 11 Joseph O’Rourke and Kenneth J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–189, 1983.
- 12 Subhash Suri. A linear time algorithm with minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.*, 35(1):99–110, July 1986. URL: [http://dx.doi.org/10.1016/0734-189X\(86\)90127-1](http://dx.doi.org/10.1016/0734-189X(86)90127-1), doi:10.1016/0734-189X(86)90127-1.
- 13 Subhash Suri. *Minimum link paths in polygons and related problems*. PhD thesis, The Johns Hopkins University, Baltimore, Maryland, 1987.

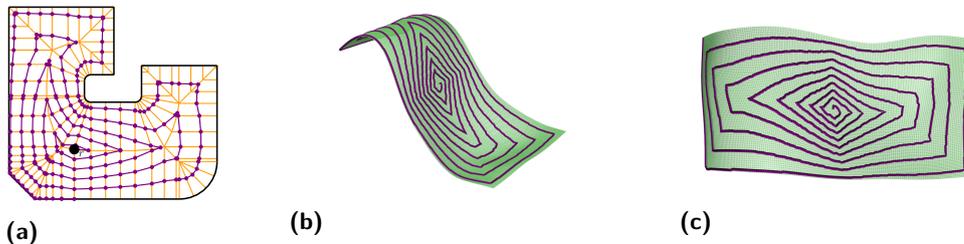
Spiral-Like Paths on Triangulated Terrains

Martin Held, Stefan de Lorenzo

Department of Computer Science, University of Salzburg
[Jakob-Haringer-Straße 2, 5020 Salzburg, Austria]
held@cs.sbg.ac.at, slorenzo@cs.sbg.ac.at

In our previous work [1] we deal with the problem of covering a planar region R (bounded by straight-line segments and circular arcs) by moving a circular disk along a continuous path: The resulting spiral-like path starts in the interior of R and ends on its boundary, and is not self-intersecting. The fundamental geometric tool for our approach is the medial axis tree $\mathcal{T}_r^*(R)$, rooted at a point r , which is formed by a combination of a discretized version of the medial axis inside R and a set of so-called clearance lines. Imagine an impulse propagating through $\mathcal{T}_r^*(R)$ which moves from r towards the leaves of $\mathcal{T}_r^*(R)$. It is possible to produce a series of consecutive wavefronts by halting this impulse at fixed points in time. The final spiral-like path consists of several laps, where one lap is a portion of the spiral that winds around r exactly once. Each lap forms a polygonal chain. Initially, the first lap L_1 and the last lap L_m are computed by interpolating between successive wavefronts. The remaining, intermediate laps are created by interpolating between L_1 and L_m , see Figure 1a. In several practical applications it is important that the minimum distance from every point on a lap to its next inner and outer lap is bounded by a user-specified distance $\Delta \in \mathbb{R}^+$. E.g., in a machining application this so-called step-over distance allows to control the material removal rate and to avoid excessive tool wear.

In the remainder of this abstract we outline ongoing work on a generalization of this path-generation strategy to spiral-like paths over piecewise-linear terrains in three dimensions. We will show that, once we have found a suitable substitute for the two-dimensional medial axis tree, the concept of impulse propagation as well as the interpolation procedure can be extended to this 3D setting. A sample spiral-like path on a simple terrain is shown in Figures 1b and 1c.



■ **Figure 1** (a) A basic spiral-like path (highlighted in purple) inside a planar region R is defined by a series of corners which are situated on the discrete medial axis tree $\mathcal{T}_r^*(R)$ (highlighted in orange). (b-c) A spiral-like path (highlighted in purple) on a triangulated terrain.

We emphasize that naïvely mapping a 2D spiral-like path onto a terrain will result in a path that lacks any distance control between neighboring laps, for every meaningful interpretation of “distance” on a terrain. However, devices (e.g., a metal detector) or humans and animals (e.g., a rescue dog) that move along such a path should be expected to have a limited range of operation. Typical distance measures that we might be interested in are the geodesic distance or the line-of-sight distance. In the sequel we explain how to compute a coverage path $\mathcal{S}(P, \Delta)$ for a given triangulated terrain P and relative to a user-defined step-over $\Delta \in \mathbb{R}^+$ such that a geodesic disk of radius Δ covers P completely when it is moved along $\mathcal{S}(P, \Delta)$. The structure of choice in our generalized approach is the geodesic Voronoi

diagram of points on a triangulated terrain. It can be defined similarly to the standard Voronoi diagram of points in the plane by replacing the Euclidean norm with the geodesic metric.

Let P be a triangulated terrain. We place point sites along the polygonal boundary ∂P of P such that a roughly uniform spacing is achieved. The resulting geodesic Voronoi diagram $\mathcal{GVD}(P)$ forms a tree $\mathcal{T}_r(P)$ rooted at a point $r \in \mathcal{GVD}(P)$, where r is height-balanced (see [2] for further details). We will refer to a path along $\mathcal{T}_r(P)$ from r to a leaf of $\mathcal{T}_r(P)$ as a source branch, with source branches of maximal length being called radial paths. It is assumed that the order of these source branches is defined by the sequence in which the corresponding leaves appear when ∂P is traversed counter-clockwise. If p and q are two points on $\mathcal{T}_r(P)$, then the (geodesic) length $\ell(p, q)$ corresponds to the length of the unique path from p to q along $\mathcal{T}_r(P)$. The geodesic height of a point p is defined by $h(p) := \max_q \ell(p, q)$, where the maximum is taken over all nodes q of the sub-tree(s) of $\mathcal{T}_r(P)$ rooted at p . Now imagine an impulse propagating through $\mathcal{T}_r(P)$ which starts at r at time $t = 0$, splits at the nodes of $\mathcal{T}_r(P)$, and discharges simultaneously at the leaves at time $t = 1$. The impulse reaches a specific point p on the radial paths of $\mathcal{T}_r(P)$ at the activation time $t_p = \frac{h(r) - h(p)}{h(r)}$. This observation can be used to assign an activation time to every point on $\mathcal{T}_r(P)$ by recursively “peeling off” the corresponding radial paths. Due to space considerations, we refer to [1] for a more detailed description of this process in the 2D setting. As time progresses the impulse covers an increasing portion of $\mathcal{T}_r(P)$. Therefore, we can construct a series of $m + 1$ uniformly spaced wavefronts by defining a uniform decomposition of time. Each wavefront is given by a series of vertices (in which the first and last vertex coincide) that are situated on consecutive source branches. These wavefronts have to be chosen carefully such that the (symmetric) Hausdorff distance $H(w_i, w_{i+1})$, under the geodesic metric, between two neighboring wavefronts w_i and w_{i+1} , with $i \in \{0, 1, \dots, m - 1\}$, is bounded by Δ .

The final spiral-like path $\mathcal{S}(P, \Delta)$ consists of a series m laps L_1, L_2, \dots, L_m . Each lap is defined by a sequence of vertices which, again, lie on the source branches. Initially, two laps are computed. The first lap L_1 is generated by gradually moving its vertices from the corresponding vertices of w_0 (i.e. r) towards the vertices of w_1 along the consecutive source branches. Similarly, the last lap L_m is created by interpolating between w_{m-1} and w_m (i.e. ∂P). Every vertex of L_1 (or L_m) is at most Δ away from r (or ∂P). (Recall that the construction of the wavefronts ensures that w_1 is at most Δ away from r and the distance between w_{m-1} and ∂P is also bounded by Δ .) To construct the remaining laps L_2, L_3, \dots, L_{m-1} , a modified impulse is used. This time it starts at the vertices of L_1 , moves along $\mathcal{T}_r(P)$, and discharges concurrently at the vertices of L_m . The vertices of the lap L_{i+1} are produced by halting this modified impulse at $t = \frac{i}{m-2}$, with $i \in \{1, 2, \dots, m - 2\}$. These laps L_2, L_3, \dots, L_{m-1} split the paths from the vertices of L_1 towards the respective vertices of L_m (along $\mathcal{T}_r(P)$) into $m - 1$ portions with a length of at most Δ .

This construction ensures that the maximum geodesic distance between neighboring wavefronts obeys the user-specified step-over Δ . We can also establish this property for specific vertices of the spiral-like path, and are investigating how to extend our distance considerations such that Δ is guaranteed to be respected along the entire path. We note that restricting our attention to a convex terrain does not seem to make the analysis any simpler.

References

- 1 Martin Held and Stefan de Lorenzo. On the Generation of Spiral-Like Paths Within Planar Shapes. *Journal of Computational Design and Engineering*, 2018, to appear.
- 2 Martin Held and Christian Spielberger. Improved Spiral High-Speed Machining of Multiply-Connected Pockets. *Computer-Aided Design and Applications*, 11(3):346–357, 2014.

Enumerating Maximal Regional Failures of Backbone Communication Networks in Near Linear Parametric Time

Balázs Vass

MTA-BME Future Internet Research Group, Budapest University of Technology and Economics
vb@tmit.bme.hu

 <https://orcid.org/0000-0002-8589-7165>

János Tapolcai¹

MTA-BME Future Internet Research Group, Budapest University of Technology and Economics
tapolcai@tmit.bme.hu

 <https://orcid.org/0000-0002-3512-9504>

Abstract

Protecting Internet backbone topologies against regional failures is becoming a fundamental problem of our society. Based on computational geometric tools, we provide a fast algorithm for detecting link sets having a significant probability of failing together.

2012 ACM Subject Classification Computational geometry, Mesh networks, Failure recovery, maintenance and self-repair

Keywords and phrases Sweep line algorithm, disaster protection, Shared Risk Link Group

1 Introduction and Model

Internet backbone networks can recover from failures that listed as *Shared Risk Link Groups (SRLGs)*. An SRLG consists of a set of links which are considered to have a significant probability of failing simultaneously. Networks can be protected against natural disasters (e.g. earthquakes) if we can list them as SRLGs.

We model the network as an undirected geometric graph $G(V, E)$ embedded in the plane \mathbb{R}^2 . Let $n := |V|$, $m := |E|$. We identify the nodes with their coordinates. The edges are considered to be line segments between their endpoints. We assume the disasters causing regional failures have a shape of a circular disk with a given radius $r \in \mathbb{R}^+$. The output of the problem is set M_r of the inclusion-wise maximal edge sets which can be intersected by a disk with radius r . We improve the existing algorithm for computing M_r offered in [2].

2 Algorithm for Determining Maximal Failures

Let X denote the set of points in $\mathbb{R}^2 \setminus V$ where at least two edges of E are intersecting. Let $x := |X|$.

► **Lemma 1** (Theorem 4 of [2]). *For every link set $M \in M_r$ there exists a disk c of radius at most r intersecting M lying in a closed disk with radius $3r$ having centre point in $V \cup X$.*

The former lemma can be proven using basic geometry/trigonometry.

► **Lemma 2** (Claim 2 of [2]). *The number of edges in G is $O(n + x)$.*

¹ The research is partially supported by the Hungarian Scientific Research Fund (OTKA K 124171).

Proof. Let $G'(V \cup X, E')$ be the planar graph obtained from dividing the edges of G at the crossings. Since every crossing enlarges the number of edges at least with two, $|E'| \geq m + 2x$. On the other hand, based on Euler's formula, $|E'| \leq 3(n + x) - 6$ since G' is planar. Thus $m \leq |E'| - 2x \leq 3n + x - 6$. ◀

► **Theorem 3** (Rephrasing part of Theorem 2.4 of [1] based on Lemma 2). *Every point of X can be reported in $O((n + x) \log(n + x))$.*

The former theorem can be proved using a classical sweep line algorithm.

For a $p \in V \cup X$ let $E(p, r)$ denote the set of edges $e \in E$ such that the distance $d(e, p) \leq 3r$. Let E_r be obtained by elongating the edges of E in both directions by $3r$. Let $G_r := (V, E_r)$. Let X_r denote the set of points in $\mathbb{R}^2 \setminus V$ where at least two edges for E_r are intersecting. Let $x_r := |X_r|$. Let ρ_r be the maximum number of edges of E intersected by a disk with radius at most r .

► **Theorem 4.** *For all $p \in V \cup X$, $E(p, r)$ can be determined in $O((n + x_r) + (n + x)\rho_r)$.*

Unfortunately the proof of Thm. 4 would exceed the limits of this Abstract. A key idea is to apply sweep line algorithm in both the vertical and horizontal directions.

For every $p \in V \cup X$, let M_r^p be the set of maximal subsets of $E(p, r)$ which can be intersected by a disk with radius at most r . Let ρ_r be the maximal cardinality of sets M_r^p .

► **Lemma 5** (Corollary of Thm 1. of [2]). *For every $p \in V \cup X$, known $E(p, r)$, M_r^p can be calculated in $O(\rho_r^5)$ and $|M_r^p|$ is $O(\rho_r^2)$.*

As presented in [2], the key idea of the lemma's proof is that there are $O(\rho_r^2)$ possible locations to check for possible maximal elements of M_r^p . In addition, the resulting candidate link sets can be pairwise compared in $O(\rho_r)$

For every $e \in E$, let $\Theta_{e,r}$ and $\theta_{e,r}$ denote the set and number of points $p \in V \cup X$ for which $e \in E(p, r)$. Let d_E be the square mean of the numbers θ_e .

► **Theorem 6.** *Given M_r^p for all $p \in V \cup X$, M_r can be calculated in $O((n + x)d_E\rho_r^5)$.*

For each edge $e \in E$, only lists M_r^p for $p \in \Theta_e$ have to be merged. Sets M_r^p , which can be determined during the previous steps of the algorithm without increasing of the complexity.

► **Theorem 7.** *M_r can be calculated in $O((n + x)d_{E,r}\rho_r^5 + (n + x_r) \log(n + x_r))$.*

Proof. Based of Theorems 3, 4 and 6, and Lemma 5, one can determine M_r in the proposed complexity by determining X , then for all $p \in V \cup X$, determining $E(p, r)$ and M_r^p , finally merging these lists into M_r . ◀

► **Corollary 8.** *If $\rho_r = O(r)$, $x = O(n)$, $x_r = O(nr)$, $\theta_{E,r} = O(r^2)$, $|M_r|$ can be calculated in $O(nr^7 + (nr) \log(nr))$.*

Note that for typical backbone topologies, Cor. 8 reflects realistic assumptions. The algorithm for determining M_r in [2] computes M_r in $O(n^2r^5)$ time adopting the assumptions of Cor. 8.

References

- 1 Mark de Berg and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 1997.
- 2 János Tapolcai, Lajos Rónyai, Balázs Vass, and László Gyimóthi. List of shared risk link groups representing regional failures with limited size. In *Proc. IEEE INFOCOM*, Atlanta, USA, may 2017.

NP-completeness of Planar Steiner Orientation

Moritz Beck

Julius-Maximilians-Universität Würzburg, Germany
beck@informatik.uni-wuerzburg.de

Johannes Blum

Julius-Maximilians-Universität Würzburg, Germany
blum@informatik.uni-wuerzburg.de

Myroslav Kryven

Julius-Maximilians-Universität Würzburg, Germany
kryven@informatik.uni-wuerzburg.de

Andre Löffler

Julius-Maximilians-Universität Würzburg, Germany
loeffler@informatik.uni-wuerzburg.de

Johannes Zink

Julius-Maximilians-Universität Würzburg, Germany
zink@informatik.uni-wuerzburg.de

1 Introduction

The STEINER ORIENTATION problem is defined as follows: given a mixed graph $G = (V, E \cup A)$ with both undirected edges E and directed arcs A and a set $T \subseteq V \times V$ of k terminal pairs, is there an orientation of all edges in E , such that for every terminal pair $(s, t) \in T$ there is an s - t -path in the resulting directed graph?

In general, this problem was shown to be NP -complete by Arkin and Hassin [1]. Cygan et al. [3] gave an $n^{O(k)}$ -time algorithm, showing it is in XP in k . Pilipczuk and Wahlström [6] improved the hardness result showing it to be $W[1]$ -hard in k . For $A = \emptyset$, however, Hassin and Megiddo [5] give a polynomial time algorithm. This raises the following question: how do restrictions on G influence complexity of STEINER ORIENTATION? These hardness proofs utilize non-planar instances. Chitnis and Feldmann [2] showed that under the Exponential Time Hypothesis, STEINER ORIENTATION cannot be solved in $f(k) \cdot n^{o(k)}$ time, even when restricting to graphs of genus 1.

In this work, we consider the PLANAR STEINER ORIENTATION problem where G is a planar graph. As a first result on computational complexity, we show the following:

► **Theorem 1.** PLANAR STEINER ORIENTATION is NP -complete.

2 Hardness Proof

To prove Theorem 1, we give a reduction from PLANAR MONOTONE 3-SAT, introduced by de Berg et al. [4] and known to be NP -complete. We use different gadgets for variables, clauses and edges. These are stitched together at shared undirected edges. Given a planar monotone 3-SAT formula F , we use these gadgets to create an instance of PLANAR STEINER ORIENTATION resembling the incidence graph of F with $|T|$ polynomial in $|F|$. Without loss of generality we assume that every variable of F occurs both negated and unnegated.

Figure 1a shows a *flip gadget*, a building block used in other gadgets. It contains two terminal pairs (s_1, t_1) and (s_2, t_2) and two undirected (red) edges. Connecting both pairs will result in opposing directions for the two undirected edges.

For every variable x in F , we have a *variable gadget* (Figure 1b). It mimics the flip gadget, providing an undirected edge e_C^x for every positive/negative clause C containing x above/below the pairs respectively. We say that the gadget is (false) true if the undirected edges are oriented (counter-)clockwise. No other orientation allows connecting both pairs.

We use two stacked flip gadgets as *edge gadgets*: by reversing direction twice, we synchronize the outer red edges. We attach these edges to a variable and a clause gadget.

For every clause C , we have a *clause gadget* (Figure 1c). It contains a terminal pair (s, t) and has an undirected edge \bar{e}_C^w for each variable w it contains. The undirected edge \bar{e}_C^y in the middle is flipped to get a consistent orientation for variables set to true. The edges f and g are synchronized by two flip gadgets to ensure that at most one of them is used to connect (s, t) . For the clause gadgets, we get the following Lemma:

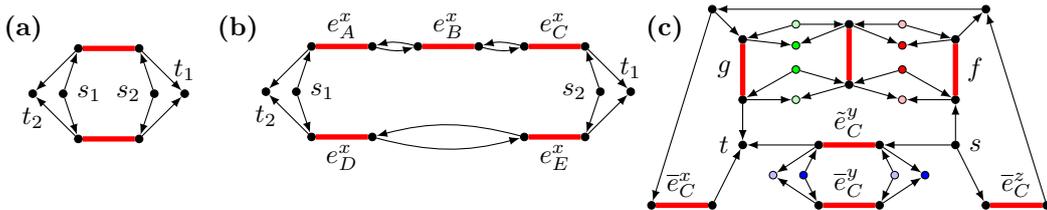
► **Lemma 2.** *All pairs of clause gadgets are connected iff ≥ 1 edge \bar{e}_C is directed to the right.*

Proof. In our construction all gadgets are self-contained and the terminal pairs of the flip gadgets can be connected. The edges f and g are both directed upwards or downwards. Hence it suffices to show the equivalence for the pair (s, t) . “ \Leftarrow ”: Case 1: If \bar{e}_C^x is directed to the right, orient the edge f away from s . Case 2: If \bar{e}_C^y is directed to the right, \tilde{e}_C^y is directed to the left. Case 3: If \bar{e}_C^z is directed to the right, orient the edge g pointing to t . In each of these cases s is connected to t . “ \Rightarrow ”: By contraposition. As we move away from s , we can neither use the edge \tilde{e}_C^y nor \bar{e}_C^z . Thus we have to use f which means it points away from s . To come to t we have to use one of the edges \bar{e}_C^x or g . This is impossible. ◀

Provided that all terminal pairs of all variable and edge gadgets are connected (which can always be achieved), the terminal pairs of all clause gadgets can be connected if and only if the formula is satisfiable. Thus, the PLANAR STEINER ORIENTATION instance has a solution if and only if the corresponding PLANAR MONOTONE 3-SAT formula is satisfiable.

Future work could involve proving $W[1]$ -hardness or looking for approximation algorithms. Other graph classes – with different geometric restrictions – could also be considered.

Acknowledgement We would like to thank Andreas Feldmann who introduced us to the problem. Parts of this work were done while the authors visited HOMONOLO 2017.



■ **Figure 1** (a) The flip gadget, used to construct edge gadgets; (b) a variable gadget with three positive and two negative occurrences; (c) a clause gadget (unlabeled (s, t) -pairs color-coded).

References

- 1 Esther M. Arkin and Refael Hassin. A note on orientations of mixed graphs. *Discrete Applied Mathematics*, 116(3):271–278, 2002.
- 2 Rajesh Chitnis and Andreas Emil Feldmann. A tight lower bound for steiner orientation. In *Proceedings of the 13th International Computer Science Symposium in Russia (CSR)*, volume 10846 of *Lecture Notes in Computer Science*, pages 65–77. Springer, 2018.

- 3 Marek Cygan, Guy Kortsarz, and Zeev Nutov. Steiner Forest Orientation Problems. *SIAM Journal on Discrete Mathematics*, 27(3):1503–1513, 2013.
- 4 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry & Applications*, 22(3):187–206, 2012.
- 5 Refael Hassin and Nimrod Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114:589–602, 1989.
- 6 Marcin Pilipczuk and Magnus Wahlström. Directed multicut is $W[1]$ -hard, even for four terminal pairs. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1167–1178. SIAM, 2016.

Blockers for Simple Hamiltonian Paths in Convex Geometric Graphs of Odd Order

Chaya Keller¹

Department of Mathematics, Ben Gurion University of the NEGEV
Be'er Sheva, Israel
kellerc@math.bgu.ac.il

Micha A. Perles

Institute of Mathematics, Hebrew University
Jerusalem, Israel
perles@math.huji.ac.il

A convex geometric graph G is a graph whose vertices are points in convex position in the plane, and whose edges are segments connecting pairs of vertices.

► **Definition 1.** Let \mathcal{F} be a family of subgraphs of G . A set of edges in $E(G)$ is called a blocking set for \mathcal{F} if it contains an edge of every element of \mathcal{F} . A *blocker* for \mathcal{F} is a blocking set of smallest possible size. The family of blockers for \mathcal{F} is denoted $\mathcal{B}(\mathcal{F})$.

Let $G = CK(n)$ be the complete convex geometric graph on n vertices. Finding the size of the blockers $\mathcal{B}(\mathcal{F})$ for a family \mathcal{F} is a natural Turán-type question, as it is clearly equivalent to the question: what is the maximal possible number of edges in a convex geometric graph on n vertices that does not include any element of \mathcal{F} as a subgraph? This question was extensively studied with respect to various families \mathcal{F} , e.g., all sets of k disjoint edges [6] and all sets of k pairwise crossing edges ([2], and see also [1]). The geometric nature of the question comes from the geometric definition of the ‘blocked’ family.

In various cases, including the aforementioned cases, the size of the blockers is known. The next natural step is to provide a *characterization* of the blockers for \mathcal{F} .

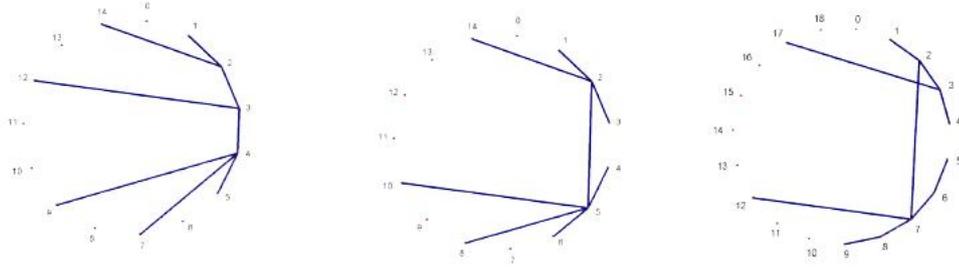
In a series of previous works, complete characterizations of blockers for several families were provided. These families include simple (i.e., non-crossing) perfect matchings (SPMs) in $G = CK(2m)$ (see [3]), triangulations in $G = CK(2m)$ (see [7]), and simple spanning trees in general geometric graphs (see [5]). In [4] we showed that the blockers for the family \mathcal{H} of simple Hamiltonian paths (SHPs) in $CK(2m)$ are exactly the same as the blockers for SPMs, i.e., simple caterpillars with a spine on the boundary of $\text{conv}(V(G))$. (A caterpillar is a tree in which all the vertices are within distance 1 of a central path. Such a path is called a spine of the caterpillar.)

In this work we consider the (much more complicated) case of SHPs in a complete convex geometric graph of odd order $2m - 1$. Our main result is a complete description of the blockers for \mathcal{H} . This time, not all blockers are caterpillar trees, and they do not even have to be simple. In order to describe the blockers, we need a few additional notions.

Let V be the set of vertices of $G = CK(2m - 1)$ (viewed as the vertex set of a convex polygon P in the plane), labelled cyclically from 0 to $2m - 2$. We denote an edge in G between vertices i, j by $[i, j]$, and a path of length r by $\langle i_1, i_2, \dots, i_r \rangle$. The edges that belong to the boundary of P (i.e., the edges of the form $[i, i + 1]$ ($0 \leq i \leq 2m - 3$), along with the edge $[2m - 2, 0]$) are called *boundary edges* of G . The *direction* of an edge $[i, j]$ is defined to be $i + j \pmod{2m - 1}$. Two edges are *parallel* if they have the same direction.

Our main theorem is the following:

¹ The work of the first author was partially supported by Grant 635/16 from the Israel Science Foundation, by the Shulamit Aloni Post-Doctoral Fellowship of the Israeli Ministry of Science and Technology and by the Kreitman Foundation Post-Doctoral Fellowship.



■ **Figure 1** Three blockers for SHPs in $CK(15)$. The left blocker is of Class A, with parameters $m = 8, \alpha = 1, \delta = 3, \epsilon_1 = 1, (\xi_1, \xi_2, \xi_3) = (4, 2, 1)$. The middle blocker is of Class B, with parameters $(m = 8, \alpha, \beta, \gamma, \delta) = (1, 2, 2, 2), \eta = 1, \epsilon_1 = 1$, and $(\xi_1, \xi_2) = (2, 1)$. The right blocker is a non-simple blocker of Class B with parameters $m = 10, (\alpha, \beta, \gamma, \delta) = (1, 3, 4, 1), \eta = 2, \epsilon_1 = 2$, and $\xi_1 = 2$.

Theorem. Let $G = CK(2m - 1)$, and let \mathcal{H} be the family of simple Hamiltonian paths in $E(G)$. Any blocker for \mathcal{H} consists of m edges, and up to cyclical rotation by $0 \leq k \leq 2m - 2$, it has one of the following two forms. Moreover, all the sets described below (Class A and Class B) are indeed blockers.

Class A. Blockers with a consecutive boundary path. The edges of the blocker are parallel to the boundary edges $[0, 1], [1, 2], \dots, [m - 1, m]$. They consist of three parts:

- (1). The boundary path $BP = \langle \alpha, \alpha + 1, \dots, m - \delta \rangle$, for some $\alpha, \delta \geq 0$ with $0 \leq \alpha + \delta \leq m - 2$.
- (2). The edges $u_i = [i - 1 - \epsilon_i, i + \epsilon_i], 1 \leq i \leq \alpha$ (where indices are taken modulo $2m - 1$), for $\epsilon_1 > \epsilon_2 \dots > \epsilon_\alpha > 0, \alpha - i + 1 \leq \epsilon_i \leq m - \delta - i - 1$. (These are the edges parallel to $[0, 1], [1, 2], \dots, [\alpha - 1, \alpha]$).
- (3). The edges $v_j = [m - j - \xi_j, m - j + 1 + \xi_j], 1 \leq j \leq \delta$, for $\xi_1 > \xi_2 \dots > \xi_\delta > 0, \delta + 1 - j \leq \xi_j \leq m - j - \alpha - 1$. (These are the edges parallel to $[m - \delta, m - \delta + 1], \dots, [m - 1, m]$), where in addition it is required that $\epsilon_1 + \xi_1 \leq m - 2$ (which means that all edges of the second part lie ‘above’ all edges of the third part).

An example of a blocker of this class is presented in the left part of Figure 1.

Class B. Blockers with a broken boundary path. The edges of the blocker are parallel to the boundary edges $[0, 1], [1, 2], \dots, [m - 1, m]$. They consist of five parts as follows:

- (1)–(2). The boundary paths $\langle \alpha, \alpha + 1, \dots, \alpha + \beta \rangle$ and $\langle \alpha + \beta + 1, \alpha + \beta + 2, \dots, m - \delta \rangle$ of lengths β, γ , respectively, for some $\alpha, \delta \geq 0, \beta, \gamma \geq 2$ with $\alpha + \delta \leq m - 5$ and $\beta + \gamma = m - \alpha - \delta - 1$. (That is, the boundary path $\langle \alpha, \alpha + 1, \dots, m - \delta \rangle$ is missing a single edge $[\alpha + \beta, \alpha + \beta + 1]$.)
- (3). The edge $[\alpha + \beta - \eta, \alpha + \beta + 1 + \eta]$, for some $1 \leq \eta \leq \min(\beta - 1, \gamma - 1)$ (which is parallel to the ‘missing’ boundary edge $[\alpha + \beta, \alpha + \beta + 1]$).
- (4). The edges $[i - 1 - \epsilon_i, i + \epsilon_i], 1 \leq i \leq \alpha$, for $\alpha + \beta - 1 > \epsilon_1 > \epsilon_2 \dots > \epsilon_\alpha > 0$. (These are the edges parallel to $[0, 1], [1, 2], \dots, [\alpha - 1, \alpha]$; we have the additional condition that they lie ‘above’ the missing boundary edge).
- (5). The edges $[m - j - \xi_j, m - j + 1 + \xi_j], 1 \leq j \leq \delta$, for $\gamma + \delta - 1 > \xi_1 > \xi_2 \dots > \xi_\delta > 0$. (These are the edges parallel to $[m - \delta, m - \delta + 1], \dots, [m - 1, m]$; we have the additional condition that they lie ‘below’ the missing boundary edge).

Blockers of this class are not caterpillars, and moreover, they are not necessarily simple. An example of a blocker of this class is presented in the middle part of Figure 1, and another example – which is not even simple – is presented in the right part of Figure 1.

References

- 1 P. Brass, G. Károlyi, and P. Valtr, A Turán-type extremal theory of convex geometric graphs, *Discrete and Computational Geometry, Algorithms Combin.*, 25, Springer, Berlin, 2003, pp. 275–300.
- 2 V. Capoyleas and J. Pach, A Turán-type theorem on chords of convex polygons, *J. Combin. Th. Ser. B.*, **56** (1992), pp. 9–15.
- 3 C. Keller and M. A. Perles, On the smallest sets blocking simple perfect matchings in a convex geometric graph, *Israel J. Math.* **187** (2012), pp. 465–484.
- 4 C. Keller and M. A. Perles, Blockers for simple Hamiltonian paths in convex geometric graphs of even order, *Disc. Comput. Geom.*, to appear.
- 5 C. Keller, M. A. Perles, E. Rivera-Campo and V. Urrutia-Galicia, Blockers for non-crossing spanning trees in complete geometric graphs, *J. Pach (ed.), Thirty Essays on Geometric Graph Theory, Springer-Verlag*, (2013) pp. 383-398.
- 6 Y.S. Kupitz, Extremal problems of combinatorial geometry, Aarhus University Lecture Notes Series **53** (1979).
- 7 C. Keller and Y. Stein, Blockers for triangulations of a convex polygon and a geometric Maker-Breaker game, submitted, 2017. Available at: <https://arxiv.org/abs/1801.00324>.

On d -D Polycubes with Small Perimeter Defect

Andrei Asinowski¹

Inst. of Discrete Mathematics and Geometry, Vienna Univ. of Technology
1040 Vienna, Austria
andrei.asinowski@tuwien.ac.at

Gill Barequet²

Yufei Zheng²

Dept. of Computer Science, The Technion—Israel Inst. of Technology,
Haifa 3200003, Israel
{barequet,yufei}@cs.technion.ac.il

Abstract

A d -dimensional polycube is a face-connected set of cells on \mathbb{Z}^d . To-date, no formulae enumerating d -dimensional polycubes by volume (number of cubes) or perimeter (number of empty cubes neighboring the polycube) are known. We investigate formulae enumerating by volume and dimension polycubes with a fixed deviation from the maximum possible perimeter.

1 Introduction

A d -dimensional *polycube* of volume n is a connected set of n cubes on \mathbb{Z}^d , where connectivity is through $(d-1)$ -dimensional faces. Two *fixed* polycubes are considered identical if one can be translated into the other; we consider here only fixed polycubes. The study of polycubes began in the 1950s in statistical physics [3] and in the 1960s in enumerative combinatorics [4].

The *perimeter* of a polycube P is defined as the number of empty cells neighboring P . Luther and Mertens [5] derive formulae enumerating n -cell polycubes (with perimeter p) that span $n-1$ or $n-2$ dimensions. For any d -D polycube P of volume n and perimeter p , it is easy to see that p is at most $M_n = 2dn - 2(n-1) = 2(d-1)n + 2$. Let $k = M_n - p$ be the “perimeter defect” of P , and $B(n, k, d)$ be the number of d -D polycubes of volume n and defect k . We show that for fixed d and $k \geq 0$, the generating function of $(B(n, k, d))$ is rational. This work generalizes our previous works [1, 2], in which we investigated 2- and 3-dimensional polycubes with a fixed defect.

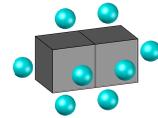


Figure 1
A 3D polycube of size 2 with perimeter 10

2 Formulae for Small Defect

To better employ the nature of defect using graph terminology, we mostly consider the dual graph P^* of a polycube P . (That is, P^* is the cell-adjacency graph of P .) With a slight abuse of notation, we shall sometimes consider P as a graph and refer to the degree of a cell of P , the number of edges in P , etc.

The *excess* of a perimeter cell X of P is the number of neighbors of X that belong to P , minus 1. An *excess cell* is a perimeter cell with non-zero excess, *i.e.*, an empty cell with at least two occupied neighboring cells. Let e be the total excess of P , *i.e.*, the sum of excesses over all perimeter cells of P . Finally, let r be the *circuit rank* of P^* (a.k.a. the *cyclomatic*

¹ Work on this paper by the first author has been supported by FWF Grant S50-N15 (Austria) in the framework of the Special Research Program “Algorithmic and Enumerative Combinatorics.”

² Work on this paper by the second and third authors has been supported in part by Grant 575/15 from the Israel Science Foundation (ISF).

number of P), i.e., the minimal number of edges that must be removed from P^* in order to make it a tree. In a connected graph $G = (V, E)$, we have $r = |E| - |V| + 1$.

► **Proposition 2.1.** For every d -dimensional polycube of volume n and perimeter p , we have (1) $p \leq 2(d - 1)n + 2$; and (2) $k = e + 2r$. ◀

The upper bound in Prop. 2.1(1) is tight. Prop. 2.1(2) helps us to identify all possible shapes of polycubes with defect k . Note that patterns of defect k can span at most $k+1$ dimensions. Hence, in order to compute $B(n, k, d)$, we need to sum up formulae for patterns that span $1 \leq \delta \leq k + 1$ dimensions, and multiply each formula of a δ -D pattern by $\binom{d}{\delta}$. This provides our general framework for deriving $B(n, k, d) = \sum_{\delta=1}^{k+1} \binom{d}{\delta} [\sum_{\pi \text{ in } \delta\text{-D}} \text{Formula}(\pi)]$. By applying this framework to small values of k , we obtain the following formulae.

k	C.P.
0	$x - 1$
1	$(x - 1)^2$
2	$(x - 1)^3$
3	$(x - 1)^4(x + 1)^2$

► **Table 1** Characteristic polynomials of 3-dimensional polycubes

► **Proposition 2.2.**

- (1) $B(n, 0, d) = d$ for $n \geq 2$ (and 1 for $n = 1$);
- (2) $B(n, 1, d) = \binom{d}{2} \cdot 4(n - 2)$ for $n \geq 2$ (and 0 for $n = 1, 2$).
- (3) $B(n, 2, d) = \binom{d}{2} [4\binom{n-2}{2} + 4\binom{n-4}{2} + 4\binom{n-2}{2} + 8] + \binom{d}{3} 3! \cdot 2^3 \binom{n-4}{2}$ for $n \geq 6$.
- (4) $B(n, 3, d) = \binom{d}{2} \left[-\frac{2091}{2} - \frac{21(-1)^n}{2} + \left(\frac{1947}{4} + \frac{5(-1)^n}{4} \right) n - 89n^2 + \frac{13}{2}n^3 \right] + \binom{d}{3} [-2328 + 1460n - 356n^2 + 32n^3] + \binom{d}{4} [-63360 + 19136n - 1920n^2 + 64n^3]$ for $n \geq 12$.

3 General Form

► **Theorem 3.1.** For any fixed dimension d , $\sum_{n \geq 0} B(n, k, d)x^n$, the generating function that enumerates polycubes with a fixed defect k with respect to their volume, is rational. Moreover, its denominator is a product of cyclotomic polynomials.

Proof. The proof consists of the following three steps.

1. Partition the set of polycubes with defect k into mutually-disjoint “pattern classes.”
2. Show that the generating function of each pattern class has the form stated in the theorem;
3. Prove that the number of pattern classes is bounded.

Then, the combination of Steps (2) and (3) implies the claim. ◀

A possible direction for future study might be to look into the general behavior of $B(n, k, d)$, where k is no longer a small constant, but rather a function of n .

References

- 1 Andrei Asinowski, Gill Barequet, and Yufei Zheng. Enumerating polyominoes with fixed perimeter defect. *Electronic Notes in Discrete Mathematics*, 61:61–67, 2017.
- 2 Andrei Asinowski, Gill Barequet, and Yufei Zheng. Polycubes with small perimeter defect. In *Proc. 29th Ann. ACM-SIAM Symp. on Discrete Algorithms, New Orleans, LA*, pages 93–100, January 2018.
- 3 Simon R. Broadbent and John M. Hammersley. Percolation processes: I. crystals and mazes. In *Mathematical Proc. of the Cambridge Philosophical Society*, volume 53, pages 629–641. Cambridge University Press, 1957.
- 4 Solomon Golomb. *Polyominoes*. C. Scribner’s Sons, 1965. 2nd edition, 1994.
- 5 Sebastian Luther and Stephan Mertens. The perimeter of proper polycubes. *J. of Integer Sequences*, 20(2):3, 2017.