

Computational Geometry

Maarten Löffler^{*1}, Eunjin Oh^{*2}, Jeff M. Phillips^{*3}, and
Alexandra Weinberger^{†4}

1 Utrecht University, NL. m.loffler@uu.nl

2 POSTECH – Pohang, KR. eunjin.oh@postech.ac.kr

3 University of Utah – Salt Lake City, US. jeffp@cs.utah.edu

4 FernUniversität in Hagen, DE. alexandra.weinberger@fernuni-hagen.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25201 “Computational Geometry”. The seminar program spanned the days from 11th May to 16th May 2025, and 39 participants from various countries were on site. Recent advances in computational geometry were presented and discussed, and new challenges were identified, in particular in relation to the two themes “parameterized complexity” and “the interplay between theory and implementation”. This report collects the abstracts of the talks and the open problems presented at the seminar, an excerpt from the panel discussion, and partial progress from the active working groups.

Seminar May 11–16, 2025 – <https://www.dagstuhl.de/25201>

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Discrete mathematics

Keywords and phrases algorithms, combinatorics, complexity, geometric computing, implementation


Digital Object Identifier 10.4230/DagRep.15.5.64

1 Executive Summary

Maarten Löffler (Utrecht University, NL)

Eunjin Oh (POSTECH – Pohang, KR)

Jeff M. Phillips (University of Utah – Salt Lake City, US)

License  Creative Commons BY 4.0 International license
© Maarten Löffler, Eunjin Oh, and Jeff M. Phillips

This Dagstuhl Seminar constituted a biennial gathering of computational geometers and topologists at the Dagstuhl venue to share recent results, and further research on some of the most important problems of the time in this field. This year the seminar focused on two central challenges within computational geometry and topology: (1) the emergence of parameterized algorithms, and (2) the interaction between implementation and theory in geometry. Within the parameterized algorithms theme, two enlightening overview talks were given. The first was on parameterized techniques in computational geometry, focusing on clustering, covering, and geometric intersection graphs. The second was on parameterized complexity and its role in geometric topology, focusing on triangulated topological manifolds. The second theme on the interplay between implementation and theory covered a broader set of topics from motion planning to persistence homology to geometric software engineering to geometric topology software to non-metric high-dimensional geometries to using Haskell for

* Editor / Organizer

† Editorial Assistant / Collector



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Computational Geometry, *Dagstuhl Reports*, Vol. 15, Issue 5, pp. 64–95

Editors: Maarten Löffler, Eunjin Oh, Jeff M. Phillips, and Alexandra Weinberger



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

low-dimensional geometry. A highlight was a panel discussion on the state of the challenges of promoting implementation-driven research in a field dominated by primarily theoretical results. Finally, an open problem session promoted a variety of intriguing unsolved issues in the field; many are stated within this report. These problems occupied many of the participants throughout the week as they were interrogated in small working groups. Some research resolution during the week, and we expect others to lead to publications soon after.

Changes to the Lottery

The Computational Geometry Dagstuhl Seminar series is the longest-running series of Dagstuhl seminars, starting in 1990 with Dagstuhl Seminar 9041 (see <https://www.dagstuhl.de/9041>) and the current seminar 25201 being its 19th iteration.

In 2013, the organizers of Dagstuhl Seminar 13101 introduced a *lottery* to select participants. The reason for this was that a steadily growing number of established names in the community being invited each time limited access for more junior researchers to this series. While this is not necessarily a bad thing and is indeed a sign of a healthy and expanding community, the organizers felt this conflicted with Dagstuhl’s unique opportunity for junior and senior researchers to interact. They reported in 2013:

We believe that the lottery created space to invite younger researchers, rejuvenating the seminar, while keeping a large group of senior and well-known scholars involved. The seminar was much “younger” than in the past, and certainly more “family-friendly.” Five young children roaming the premises created an even cosier atmosphere than we are used in Dagstuhl. Without decreasing the quality of the seminar, we had a more balanced attendance than in the past. Feedback from both seminar participants and from researchers who were not selected was uniformly positive.

Since 2013, the lottery has been used each time to select most of the participants. Over time, though, the task of maintaining an accurate list of “the community” has proven challenging, and through its lack of transparency the series also acquired some mystery, specifically surrounding the questions how one enters the lottery.

This year, we have for the first time not used the hand-curated list, and instead automatically generated a list of names as input to the lottery. Here we report on how the lottery was run this year.

A large Dagstuhl Seminar has space for about about 45 participants. The organizers reserved 10 seats for hand-selected invitees based on the chosen themes for the seminar. The remaining 35 seats were selected through the lottery. The organizers have used DBLP’s SparQL interface <https://sparql.dblp.org/> to generate a list of names to enter the lottery. The query used this year was:

“all people who have at least 3 publications in SoCG, of which at least 1 appeared in the last 5 years”.

From this list, a weighted random selection was made, taking into account Dagstuhl’s requirements on geographical and gender diversity, the themes of the seminar, and invitations to previous Dagstuhl Seminars.

2 Table of Contents

Executive Summary

<i>Maarten Löffler, Eunjin Oh, and Jeff M. Phillips</i>	64
---	----

Overview of Talks

Parametrized Algorithms in Geometry <i>Sayan Bandyapadhyay</i>	68
Lessons Learned in Implementation <i>Benjamin Burton</i>	68
Better Late than Never: The Complexity of Arrangements of Polyhedra <i>Otfried Cheong</i>	69
Fast Persistence for 1D Functions (In Gudhi) <i>Marc Glisse</i>	69
The Practice and Theory of Sampling-Based Motion Planning <i>Dan Halperin</i>	70
Maximal 2-Planar Graphs: A Computational Proof for a Graph Drawing Problem <i>Michael Hoffmann</i>	70
Implementing (Geometric) Algorithms as a Less-Initiated Software Engineer <i>Gert Meijer</i>	71
Motion Planning for Introverts: Social Distancing in Various [F/N]orms <i>Valentin Polishchuk</i>	71
The Geometry of the Set of Equivalent Linear Neural Networks <i>Jonathan Shewchuk</i>	72
Parameterised Complexity in Geometric Topology <i>Jonathan Spreer</i>	72
hgeometry: Experience Report Implementing CG Algorithms <i>Frank Staals</i>	73
Google, Balls, and Technology <i>Hubert Wagner</i>	73
A Free Lunch: Manifolds of Positive Reach Can Be Smoothed Without Decreasing the Reach <i>Mathijs Wintraecken</i>	73

Panel Discussion: On the Interplay Between Implementing Geometric Algorithms and Theory	74
--	----

Open Problems and Working Groups	75
---	----

Covering by Double Disks <i>Sayan Bandyapadhyay</i>	75
Is Finding Positive Euler Characteristic Normal Surfaces FPT? <i>Benjamin Burton</i>	76
Turning Planar Triangulations Inside Out <i>Jeff Erickson and Birgit Vogtenhuber</i>	77


Morphing Long-Geodesic Embeddings on the Sphere <i>Jeff Erickson</i>	78
Approximate Simplification of Flag Filtrations <i>Marc Glisse, Jeff M. Phillips</i>	79
Sub-Alpha-Complex in High Dimension <i>Marc Glisse, Jeff M. Phillips</i>	80
Treewidth of 3-Manifolds <i>Benjamin Burton, Kristóf Huszár, Jonathan Spreer, Yelena Yuditsky</i>	81
WSPD for Low Density Graphs <i>Joachim Gudmundsson</i>	84
Consistent System of Shortest Paths <i>Francis Lazarus, Jeff Erickson, Raimund Seidel, Ling Zhou</i>	85
Graph Tiles <i>Oswin Aichholzer, Robert Ganian, Phillip Keldenich, Maarten Löffler, Gert Meijer, Alexandra Weinberger, Carola Wenk</i>	87
Approximate Minlink Paths in Subquadratic Time <i>Valentin Polishchuk, Sayan Bandyapadhyay, Mark de Berg, Joachim Gudmundsson, André Nusser, Frank Staals</i>	89
Alexander Theorem for Knotted Spheres <i>Mathijs Wintraecken and Kristóf Huszár</i>	91
Universal Point Sets <i>Alexander Wolff, Stefan Felsner, Michael Hoffmann, Maarten Löffler, Fabrizio Montecchiani, Yoshio Okamoto, Jonathan Spreer</i>	92
Participants	95

3 Overview of Talks

This section contains abstracts of talks that were given during the seminar. Sayan Bandyapadhyay and Jonathan Spreer were invited to speak on the topic of parameterized computational geometry and topology, respectively, one of the two themes of the seminar. For the theme of theoretical questions arising in implementing geometric Benjamin Burton, Marc Glisse, Danny Halperin, Gert Meijer, Frank Staals, and Hubert Wagner were invited to give a talk. Additionally, Ottfried Cheong, Michael Hoffmann, Valentin Polishchuk, Jonathan Shewchuk, Mathijs Wintraecken, gave talks in the area of computational geometry.

3.1 Parametrized Algorithms in Geometry


Sayan Bandyapadhyay (Portland State University, US)

License  Creative Commons BY 4.0 International license
© Sayan Bandyapadhyay

The design of parameterized algorithms for geometric problems has become a well-established area of research, driven in part by the many natural parameters that arise in these settings. While the field has a rich history, earlier efforts were often fragmented, with only a few problems receiving systematic attention. In contrast, recent work has adopted a more unified approach, focusing on the development of general frameworks that can address a broader class of problems. In this talk, we explore three central topics – clustering, covering, and geometric intersection graphs – where parameterized complexity has played a significant role. We review the literature in these areas, highlight key techniques, and pose several open questions for future investigation.

3.2 Lessons Learned in Implementation

Benjamin Burton (University of Queensland – Brisbane, AU)

License  Creative Commons BY 4.0 International license
© Benjamin Burton

We talk through some of the lessons learned through 26 years of working on the topological software package Regina, and from major computational projects including the tabulation of prime knots up to 20 crossings. Regina implements exact algorithms for 3-manifolds, 4-manifolds, knots and links, and is freely available from regina-normal.github.io.

3.3 Better Late than Never: The Complexity of Arrangements of Polyhedra

Otfried Cheong (Scalgo, DK)

License © Creative Commons BY 4.0 International license

© Otfried Cheong

Joint work of Boris Aronov, Sang Won Bae, Sergio Cabello, Otfried Cheong, David Eppstein, Christian Knauer, Raimund Seidel

Main reference Boris Aronov, Sang Won Bae, Sergio Cabello, Otfried Cheong, David Eppstein, Christian Knauer, Raimund Seidel: “Better Late than Never: the Complexity of Arrangements of Polyhedra”, CoRR, Vol. abs/2506.03960, 2025.

URL <https://doi.org/10.48550/ARXIV.2506.03960>

Let \mathcal{A} be the subdivision of \mathbb{R}^d induced by m convex polyhedra having n facets in total. We prove that \mathcal{A} has combinatorial complexity $O(m^{\lceil d/2 \rceil} n^{\lfloor d/2 \rfloor})$ and that this bound is tight. The bound is mentioned several times in the literature, but no proof for arbitrary dimension has been published before.

An earlier version of the paper appeared in EuroCG 2025, with a proof for general position only.

3.4 Fast Persistence for 1D Functions (In Gudhi)

Marc Glisse (INRIA – Orsay, FR)

License © Creative Commons BY 4.0 International license

© Marc Glisse

Main reference Marc Glisse: “Fast persistent homology computation for functions on \mathbb{R} ”, CoRR, Vol. abs/2301.04745, 2023.

URL <https://doi.org/10.48550/ARXIV.2301.04745>

0-dimensional persistent homology is known, from a computational point of view, as the easy case. Indeed, given a list of n edges in non-decreasing order of filtration value, one only needs a union-find data structure to keep track of the connected components and we get the persistence diagram in time $O(n\alpha(n))$. The running time is thus usually dominated by sorting the edges in $\Theta(n \log(n))$. A little-known fact is that, in the particularly simple case of studying the sublevel sets of a piecewise-linear function on \mathbb{R} or \mathbb{S}^1 , persistence can actually be computed in linear time. This talk presents a simple algorithm that achieves this complexity and an extension to image persistence. An implementation is available in Gudhi.

3.5 The Practice and Theory of Sampling-Based Motion Planning

Dan Halperin (Tel Aviv University, IL)

License © Creative Commons BY 4.0 International license
© Dan Halperin

Main reference Oren Salzman: “Sampling-based robot motion planning”, *Commun. ACM*, Vol. 62(10), pp. 54–63, 2019.

URL <https://doi.org/10.1145/3318164>

Main reference Kiril Solovey, Dan Halperin: “Sampling-Based Bottleneck Pathfinding with Applications to Fréchet Matching”, in *Proc. of the 24th Annual European Symposium on Algorithms, ESA 2016, August 22–24, 2016, Aarhus, Denmark, LIPIcs*, Vol. 57, pp. 76:1–76:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

URL <https://doi.org/10.4230/LIPICS.ESA.2016.76>

Main reference Kiril Solovey, Dan Halperin: “Efficient sampling-based bottleneck pathfinding over cost maps”, in *Proc. of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24–28, 2017*, pp. 2003–2009, IEEE, 2017.

URL <https://doi.org/10.1109/IROS.2017.8206019>

Since their introduction in the 1990s, sampling-based (SB) planners have become a central tool for solving motion planning problems, both in robotics and beyond. In this talk, I will give a brief introduction to SB planning, highlighting major milestones in its development and the types of theoretical guarantees these methods often provide. I will then discuss the application of SB planning to Fréchet-type problems, as well as its implementation in our DiscoPygal platform – a suite of tools designed for rapid development and experimentation with motion planning in 2D environments. DiscoPygal is built using Python bindings over CGAL. I will conclude with several open problems in SB planning.

3.6 Maximal 2-Planar Graphs: A Computational Proof for a Graph Drawing Problem

Michael Hoffmann (ETH Zürich, CH)

License © Creative Commons BY 4.0 International license
© Michael Hoffmann

Joint work of Michael Hoffmann, Meghana Mallik Reddy, Shengzhe Wang

Main reference Michael Hoffmann, Meghana M. Reddy: “The Number of Edges in Maximal 2-Planar Graphs”, in *Proc. of the 39th International Symposium on Computational Geometry, SoCG 2023, June 12–15, 2023, Dallas, Texas, USA, LIPIcs*, Vol. 258, pp. 39:1–39:15, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

URL <https://doi.org/10.4230/LIPICS.SOCG.2023.39>

One of the main meta questions in the study of beyond-planar graphs is: Which of the many nice properties of planar graphs carry over to a more general setting? One of these properties is sparsity and one of the most natural and well-studied classes of beyond-planar graphs is the family of k -planar graphs. A graph is k -planar if it can be drawn in the plane such that every edge has at most k crossings. It is known that every k -planar graph on n vertices has at most $c_k n$ edges, for some constant $c_k \leq 3.81\sqrt{k}$. However, several other nice properties of planar graphs do not carry over to k -planar graphs. For instance, every *maximal* planar graph (adding any edge yields a nonplanar graph) on n vertices has exactly $3n - 6$ edges. Such a correspondence does not hold for maximal k -planar graphs, with $k \geq 1$.

We studied the case $k = 2$ specifically, and could show that every maximal 2-planar graph on $n \geq 5$ vertices has at least $2n$ edges. Compared to the upper bound of $5n - 10$ from the literature, and also compared to the $3n - 6$ bound for planar graphs, this may not sound too impressive. Thus, we also tried to find examples of maximal 2-planar graphs with few

edges. Indeed, it is not too difficult to come up with some plausible candidates. Certifying 2-planarity is easy: Just describe a corresponding drawing. But proving maximality turns out to be a challenge. The reason is that – in contrast to planarity – testing if a given graph is k -planar, for $k \geq 1$, is known to be NP-complete.

Our attempts to prove maximality using pencil and paper quickly turned into a mess of case distinctions. Thus – in the spirit of the focus theme about *the interplay between theory and implementation* – it seemed much more reasonable to let a computer check all these cases. But also for a computer-based proof it is not really obvious how to proceed: Our graphs form a conceptually infinite family, where already initial slices are quite large for an exhaustive computation. In this talk I briefly discuss how we overcame these challenges so as to obtain a computational proof of the following statement: For every n there exists a maximal 2-planar graph on n vertices with $2n + c$ edges, where $c \leq 350$ is an absolute constant, independent of n .

3.7 Implementing (Geometric) Algorithms as a Less-Initiated Software Engineer

Gert Meijer (NHL Stenden Hogeschool – Leeuwarden, NL)

License © Creative Commons BY 4.0 International license
© Gert Meijer

This talk gives an overview of my experience implementing theoretical algorithms in an applied software engineering context. This potential audience has a lower mathematical background. I present the challenges we face when trying to translate papers into ‘mortal’ language, as well as the problems that arise during the implementation phase. I advocate for more accessible papers and end with a comparison of software engineering trends. I hope these insights contribute to making theoretical work more accessible and impactful.

3.8 Motion Planning for Introverts: Social Distancing in Various [F/N]orms

Valentin Polishchuk (Linköping University, SE)

License © Creative Commons BY 4.0 International license
© Valentin Polishchuk
Joint work of Reilly Browne, Kevin A. Buchin, Omrit Filtser, Mayank Goswami, Mart Hagedoorn, Prahlad Narasimham Kasthurirangan, Joseph S. B. Mitchell, Valentin Polishchuk, Roman Voronov

Maintaining separation/invisibility for static/moving agents is the goal in various applications, motivated by security/privacy and similar considerations. I will present several recent results on the topic, ranging from introduction of a new path (dis)similarity measure to gender-aware science to resolving long-standing open problems.

Based on FoCS23, WADS25, ITCS23, SoCG20, FUN18 work with R. Browne, P. Kasthurirangan, J. Mitchell, M. Hagedoorn, O. Filtser, M. Goswami, K. Buchin, L. Sedov, R. Voronov.

3.9 The Geometry of the Set of Equivalent Linear Neural Networks

Jonathan Shewchuk (University of California – Berkeley, US)

License © Creative Commons BY 4.0 International license
© Jonathan Shewchuk

Joint work of Sagnik Bhattacharya, Jonathan Shewchuk

Main reference Jonathan Richard Shewchuk, Sagnik Bhattacharya: “The Geometry of the Set of Equivalent Linear Neural Networks”, CoRR, Vol. abs/2404.14855, 2024.

URL <https://doi.org/10.48550/ARXIV.2404.14855>

A linear neural network computes a linear transformation of its input vector. Given a fully-connected linear network, the set of all weight vectors for which the network computes the same linear transformation is an algebraic variety in weight space, called a fiber under the matrix multiplication map. Sometimes this variety a manifold, but usually not. The rank stratification of a fiber is a natural partition of the fiber into manifolds of various dimensions called strata. We characterize how these strata are connected to each other. They satisfy the frontier condition: if a stratum intersects the closure of another stratum, then the former stratum is a subset of the closure of the latter stratum. This subset relationship can be expressed as a partial order with a single minimal element, a stratum that is included in the closure of every other stratum. Our main result describes the relationship between this partial order and the ranks of certain matrices in the network. Each stratum represents a different pattern of information flow through the network, and moves up the hierarchy may offer neural network training algorithms more opportunities to succeed. To support this construction, we propose a candidate for the “fundamental subspaces of a linear neural network”, analogous to the four fundamental subspaces of a matrix.

3.10 Parameterised Complexity in Geometric Topology

Jonathan Spreer (University of Sydney, AU)

License © Creative Commons BY 4.0 International license
© Jonathan Spreer

A major goal in the field of geometric topology is the algorithmic study of topological manifolds, typically represented by triangulations. Specifically, we want to answer topological questions about manifolds by running discrete algorithms on their triangulations.

Methods from parameterised complexity play an important role in the development of these algorithms: On the one hand, parameterised hardness results precisely state how complicated triangulations can obstruct efficient solutions to algorithmic problems; on the other hand, a host of fixed-parameter tractable algorithms exist to solve hard topological problems efficiently for well-behaved manifolds and/or triangulations.

In my talk I will survey hardness results as well as practical algorithms. Some of these are implemented in the low-dimensional topology software Regina. The algorithms are fixed-parameter tractable in the first Betti number of the underlying manifold, or the treewidth of the dual graph of the input triangulation. I will then go on to survey theoretical results about how topological properties of a manifold influence the treewidth of the dual graph of its triangulations.

3.11 hgeometry: Experience Report Implementing CG Algorithms

Frank Staals (Utrecht University, NL)

License © Creative Commons BY 4.0 International license
© Frank Staals

hgeometry is a pure Haskell library implementing some textbook computational geometry algorithms “from scratch”. In the talk I will discuss some of the lessons learned while implementing these algorithms. The main points are:

- the easy things are hard (or at least annoying),
- degeneracies are a real problem,
- implementing the algorithm itself was only 1/3rd of the work,
- property testing was very useful.

Two challenges in property testing with which the theory community could help (i) are generating a good distribution of test data (that includes corner cases and degeneracies) and (ii) developing approaches for efficiently shrinking (simplifying) a data set containing a bug.

3.12 Google, Balls, and Technology

Hubert Wagner (University of Florida – Gainesville, US)

License © Creative Commons BY 4.0 International license
© Hubert Wagner

Joint work of Herbert Edelsbrunner, Hana Dal Poz Kouřimská, Tuyen Pham, Žiga Virk, Hubert Wagner

I start from a computational geometry problem arising from my work on the google alerts service (<https://www.google.com/alerts>). Specifically, nearest neighbour and ball search in spaces that are *not* metric spaces. One example is the Kullback–Leibler (KL) divergence which is used to compare pairs of discrete probability distributions (in applications, e.g. a text document is often represented as a discrete probability distribution.) This is mostly a pretext to describe certain nice properties of Bregman divergences, whose one member is the KL divergence. I show several standard geometric constructions (Delaunay triangulation, Čech filtration, k-means clustering, kd-trees) that naturally extend to the Bregman setting. Additionally, this talk serves as a starting point to a discussion about the interplay between practical applications and theory of computational geometry – which is relevant to the ongoing addition of a second, practical track to the SoCG conference.

3.13 A Free Lunch: Manifolds of Positive Reach Can Be Smoothed Without Decreasing the Reach

Mathijs Wintraecken (Centre Inria d’Université Côte d’Azur, FR)

License © Creative Commons BY 4.0 International license
© Mathijs Wintraecken

Joint work of André Lieutier, Hana Dal Poz Kouřimská, Mathijs Wintraecken

Assumptions on the reach are crucial for ensuring the correctness of many geometric and topological algorithms, including triangulation, manifold reconstruction and learning, homotopy reconstruction, and methods for estimating curvature or reach. However, these

assumptions are often coupled with the requirement that the manifold be smooth, typically at least C^2 . In this paper, we prove that any manifold with positive reach can be approximated arbitrarily well by a C^∞ manifold without significantly reducing the reach, by employing techniques from differential topology – partitions of unity and smoothing using convolution kernels. This result implies that nearly all theorems established for C^2 or manifolds with a certain reach naturally extend to manifolds with the same reach, even if they are not C^2 , for free!

4 Panel Discussion: On the Interplay Between Implementing Geometric Algorithms and Theory

Moderator: Jeff M. Phillips

Panelists: Dan Halperin, Jonathan Shewchuk, Carola Wenk, Benjamin Burton

The panel was convened to discuss the challenges in implementing geometric algorithms, and how that led to developments in the theoretical advancements and understanding. While this was in planning for over a year before the Dagstuhl Seminar, it became very timely after a large discussion on Implementation and Application focused papers at the 2024 International Symposium on Computational Geometry (SoCG), which led to a Task Force on the topic, which released a report a month before seminar.

The panel started with discussion on successes in and surprising outcomes from implementing geometric algorithms. Where did these outcomes differ from what was expected. Halperin and Burton referred to their earlier talks about how methods in robotics or enumerative topology changed the course of the field. In robotics, it was how simple sample-based motion planning made complicated geometry analysis overly detailed, and in the topology space, it was how by reimplementing (and optimizing!) worst-case expensive methods (eventually) led to really fast approaches in practice. Wenk talked about the surprise in the complexity in working with real-world GPS data, which led to eventually enough material to write a book. And Shewchuk talked about how certain lingering challenges in constrained triangulations took over a decade until only recently what seems like the “right” solution was published.

The panel was also asked about the interplay between experimental and theoretical explorations of the same topic led to advancements in both. Halperin described how the theory communities formalization of the motion planning problem paved the way of the simpler sampling-based methods. Burton mentioned that parameterized algorithms were linked to the topology problems to explain how they could run so fast, and formalize the sort of optimizations that would be possible in practice.

The discussion turned to the challenges implementation-driven work faces, how it is valued, and where it should be published. The challenges of publishing papers on the influential CGAL library were brought up. There were no definitive answers, but the proposed second track at SoCG was discussed, and those working on implementation complements to geometric work were encouraged to continue, and to submit their work for publication.


Finally, the panel was asked what lessons could be learned from their experience on the interaction between implementation and theoretical approaches to geometric problems. While Shewchuk argued that more should be taken from these interactions, Halperin ended on the note that despite it being a struggle, these different approaches had done a lot to move the field forward.

5 Open Problems and Working Groups

This section contains open problems that have been proposed on the start of the seminar. Many of the open problems have then be discussed in more depth during the week, two even resolved in full. We expect the working groups that have formed to continue their joined research on the problems beyond the seminar and potentially further collaboration will form still afterwards. A short overview of some of the ideas and discussions happening during the week has also been added to this section (to the respective open problem).

5.1 Covering by Double Disks

Sayan Bandyapadhyay (Portland State University, US)

License  Creative Commons BY 4.0 International license
© Sayan Bandyapadhyay

We are given m access points and n client points. At each access point, one can place a disk of radius 1 or 2. The goal is to pick a radius (i.e., a disk) for each access point so that (1) the chosen disks cover each client point, and (2) the number of client points covered by the radius 1 disks is maximized.

This problem is known to be NP-hard, and there are two polynomial-time approximation algorithms:

- 4-approximations [1]
- 2.5-approximations [2], FPT in sparsity and the number of radius 2 disks in an optimal solution, also $W[1]$ -hard in the number of radius 2 disks.

Questions.

1. What is the best approximation possible in polynomial time?
2. Is the problem FPT in the number of radius 1 disks in an optimal solution?

5.1.1 Progress

This problem appears to be equivalent to discrete unit disk cover (given a set of points and a set of equal-radius disks, choose the smallest set of disks that covers all the points) by the following observations:

- All clients that are within distance at most 1 from at least 1 access point will be covered anyway, so we might as well throw them away.
- All access points that are at distance larger than 2 from any remaining client point will never be a radius 2 disk, so we might as well throw them away.
- For the remaining access points, placing radius 1 disks will not cover any more clients, so all remaining clients must be covered by radius 2 disks.
- Minimizing the number of radius 2 disks now is the same as maximizing the number of radius 1 disks in the original question.

However, we are now approximating or parameterizing w.r.t. the number of non-chosen disks rather than the number of chosen disks, so known results for DUDC do not necessarily carry over.

References

- 1 Sayan Bandyapadhyay, Anil Maheshwari, Sasanka Roy, Michiel Smid, and Kasturi R. Varadarajan. Geometric covering via extraction theorem. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 287 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.7.
- 2 Sayan Bandyapadhyay and Eli Mitchell. Approximation and parameterized algorithms for covering with disks of two types of radii. In Pat Morin and Eunjin Oh, editors, *19th Algorithms and Data Structures Symposium (WADS)*, 2025. To appear.

5.2 Is Finding Positive Euler Characteristic Normal Surfaces FPT?

Benjamin Burton (University of Queensland – Brisbane, AU)

License  Creative Commons BY 4.0 International license
 Benjamin Burton

5.2.1 Problem Statement

The setting is a 3-manifold triangulation. A *normal surface* is a properly embedded surface that meets each tetrahedron in some collection of triangles and/or quadrilaterals (e.g., no tubes, no bubbles, no octagons, etc.).

The main bottleneck for unknot recognition, 3-sphere recognition, prime decomposition and some other core 3-manifold algorithms is this: find a non-trivial connected normal surface with positive Euler characteristic, or determine that no such surface exists. Here *non-trivial* means that the surface is not just a collection of triangles that build a sphere around a vertex (in other words, there is at least one quadrilateral).

Currently the best known running time is $O(3^n \cdot \text{poly}(n))$, where n is the number of tetrahedra. However, it seems plausible that this could be FPT in the treewidth of the dual graph of the triangulation. If so, this would give immediate FPT results for several important topological algorithms (e.g., those listed earlier).

Some further notes: normal surfaces are described by vectors in \mathbb{R}^{7n} (each entry counts the number of triangles or quadrilaterals in some position in some tetrahedron). These vectors must satisfy some linear equations, linear inequalities, and combinatorial constraints. The inequalities and combinatorial constraints are local to each tetrahedron, and the equations come from gluings between adjacent tetrahedra. In this sense, the structure of the equations mirrors the structure of the dual graph, which is one reason such an FPT result might be plausible.

It is also an open problem to prove that this same problem is NP-hard.

5.3 Turning Planar Triangulations Inside Out

Jeff Erickson (University of Illinois Urbana-Champaign, US) and Birgit Vogtenhuber (TU Graz, AT)

License © Creative Commons BY 4.0 International license
© Jeff Erickson and Birgit Vogtenhuber

5.3.1 Problem Statement

Let T be an arbitrary planar straight-line triangulation, with a triangular outer face, such that the origin $(0,0)$ lies in the interior of some bounded face, and no two vertices lie on the same line through the origin. An *eversion* of T is another planar straight-line triangulation satisfying three conditions:

- T and T' are embeddings of the same maximal planar graph G .
- Every vertex of G lies on the same ray from the origin in both T and T' .
- For every ray r from the origin, the sequence of edges of T that r intersects is the reversal of the sequence of edges of T' that r intersects. Equivalently, every face of T , except the outer face and the face containing the origin, has the opposite orientation of the corresponding face of T' . (The outer face of T must be the face containing the origin in T' , and vice versa.)

Prove or disprove. Every planar straight-line triangulation has an eversion.

We know how to construct an eversion of a given n -vertex triangulation T , if one exists, in $O(n^{\omega/2})$ time, by solving a certain system of linear equations. (The variables in this system are reciprocals of distances of vertices from the origin.) This is also the fastest algorithm we know to decide whether an eversion of T exists.

This problem can be viewed as a radial variant of the *hierarchical planar graph drawing* problem, which asks whether a given planar graph has a straight-line embedding where each vertex lies on a prescribed y-coordinate. This problem has a long history, culminating in a linear-time algorithm by Klemz [1].

This problem comes out of work by Erickson and Howard [3] on spherical morphing. The problem can be equivalently formulated in terms of spherical embeddings as follows: Given a shortest-path triangulation T in the open northern hemisphere, is there an isomorphic shortest-path triangulation T' in the open southern hemisphere, such that every vertex lies on the same longitude in both T and T' ? (In the language of our paper: Is every northern triangulation T , is there a θ -equivalent southern triangulation T' ?)

5.3.2 Progress

A structure that might cause complications are “rotors” or “lens-shutters”. Essentially, this is a sequence of e_0, \dots, e_k of disjoint line segments that cyclically overlap, such that for each i along the ray through of one endpoint of e_i , the intersection with $e_{i+1 \bmod k}$ lies closer to the origin than the endpoint. I’ll call this endpoint the “far endpoint” f_i of e_i . Equivalently, every edge of a rotor also has a “near endpoint” c_i . Less formally, all near endpoints are visible from the origin, but no far endpoints.

Rotors are the simplest sets of segments that do not have a *radial shelling order*. A radial shelling order indexes the segments s_1, s_2, \dots, s_n , so that if any line segment from the origin to any segment s_i crosses another segment s_j , then $j < i$. Arguments of of Awartani and Henderson [2, 3] imply that any set of segments that is radially shellable has an eversion.

An argument of Pallazi and Snoeyink [4] implies that if any ray from the origin misses all the segments, then the segments are radially shellable. Thus, rotors must surround the origin.

Further, we asked whether it might be that a rotor can be drawn with straight edges but needs a very specific shape of the polygon $F = f_0, \dots, f_k$ formed by the far endpoints. We think that we can prove the following: If there is a realization in which F is a star-shaped polygon, then there is also a realization where the vertices of F lie on a circle, or any other *convex* polygon. Likewise, there should be one for any star-shaped polygon F (with the vertices obeying the given angle condition).

It is still unclear whether a proof that every rotor can be everted would imply that any planar straight-line graph, or even every set of disjoint segments, can be everted.

References

- 1 Boris Klemz. Convex drawings of hierarchical graphs in linear time, with applications to planar graph morphing. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6–8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPICs*, pages 57:1–57:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ESA.2021.57.
- 2 Marwan Awartani and David W. Henderson. Spaces of geodesic triangulations of the sphere. *Transactions of the American Mathematical Society*, 304(2):721–732, 1987. doi:10.2307/2000738.
- 3 Jeff Erickson and Christian Howard. Shelling and sinking graphs on the sphere. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry (SoCG)*, volume 332 of *LIPICs*, pages 47:1–47:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.SoCG.2025.47.
- 4 Larry Palazzi and Jack Snoeyink. Counting and reporting red/blue segment intersections. *Graphical Models and Image Processing*, 56(4):304–310, 1994. doi:10.1006/cgip.1994.1027.

5.4 Morphing Long-Geodesic Embeddings on the Sphere

Jeff Erickson (University of Illinois Urbana-Champaign, US)

License  Creative Commons BY 4.0 International license
© Jeff Erickson

5.4.1 Problem Statement

A *geodesic* on the sphere is any arc of a great circle. A geodesic is *long* if it contains two antipodal points and *short* otherwise; every short geodesic is the unique shortest path between its endpoints. (Classically, the word “geodesic” refers only to shortest paths on the sphere; here I’m using the more general definition from Riemannian geometry.)

A *geodesic embedding* of a graph G on the sphere maps the vertices of G to distinct points and the edges of G to interior-disjoint geodesics. A geodesic embedding is *short* if each of its edges is short, and *long* otherwise.

Prove or disprove. Any two isomorphic geodesic embeddings of the same planar graph on the sphere are connected by a continuous family of geodesic embeddings.

In 1944, Cairns [1] proved that any two *short* geodesic *triangulations* of the sphere are connected by a continuous family of *short* geodesic triangulations. Cairns’s proof can be generalized to short geodesic embeddings of 3-connected planar graphs. Using ad-hoc arguments, we can show that the space of all geodesic embeddings of K_4 is connected, but nothing else appears to be known.

One significant complication is that if an edge of G is short in one embedding but long in the other, the endpoints of that edge must be *exactly* antipodal at some moment during the morph. Another complication is that specifying the motion of the vertices is not sufficient; the same vertex coordinates and rotation system can be consistent with multiple geodesic embeddings, even if no two vertices are antipodal.

Morphs are known to exist between isomorphic straight-line planar embeddings (starting with Cairns), between isotopic geodesic embeddings on the flat torus [2], and between isotopic geodesic embeddings on any surface with negative curvature [3]. In fact, for all of those surfaces, the space of all geodesic embeddings of a given graph, within each isotopy class of embeddings, modulo rigid motions on the surface, is simply connected. The only orientable surface for which this problem is open is the sphere!

References

- 1 Stewart S. Cairns. Isotopic deformations of geodesic complexes on the 2-sphere and on the plane. *Annals of Mathematics*, 45(2):207–217, 1944. URL: <http://www.jstor.org/stable/1969263>.
- 2 Erin Wolf Chambers, Jeff Erickson, Patrick Lin, and Salman Parsa. How to morph graphs on the torus. In Daniel Marx, editor, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2759–2778, 2021. doi:10.1137/1.9781611976465.164.
- 3 Yanwen Luo, Tianqi Wu, and Xiaoping Zhu. The deformation space of geodesic triangulations and generalized Tutte’s embedding theorem. *Geometry & Topology*, 7:3361–3385, 2023. doi:10.2140/gt.2023.27.3361.

5.5 Approximate Simplification of Flag Filtrations

Marc Glisse (INRIA – Orsay, FR), Jeff M. Phillips (University of Utah – Salt Lake City, US)

License © Creative Commons BY 4.0 International license
© Marc Glisse, Jeff M. Phillips

5.5.1 Problem Statement

Computing persistent homology usually takes time linear in the number of cells of the filtered complex involved. To speed things up, one then needs a small complex. The first (geometric) approach is to define a small complex to begin with. The second (combinatorial) approach, the one I am interested in today, is to simplify a filtration before computing its persistent homology. This only makes sense if we can simplify the complex without actually listing its cells explicitly (same cost as computing PH). A common case is flag filtrations (like Vietoris-Rips), which can be represented by a graph with a filtration value for each vertex and edge, and where higher dimensional cells are implicitly defined as the cliques of the graph. The goal of the simplification is then to replace the graph with a smaller graph before listing its simplices to compute PH (so there are fewer simplices to list).

Glisse and Pritam [1] provide an approach for simplifying the graph while exactly preserving its PH. However, in many cases, we would be happy with an approximation.

Question. How can we simplify the graph, more than in the exact case, but still with guarantees on the error?

5.5.2 Progress

The following approaches and questions arose during the seminar.

- Forcibly reintroduce some geometry, embedding in $\dim \log n$.
- Could rounding to a grid be more efficient than what the paper does, sometimes, because it gives some edges the same value and thus helps jump over them? Needs experiments; but initial experiments showed this did not provide an improvement.
- Simplification with Morse matching (not at the graph level, just a first step to understand what is possible?)

References

- 1 Marc Glisse and Siddharth Pritam. Swap, shift and trim to edge collapse a filtration. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPICs*, pages 44:1–44:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SOCG.2022.44.

5.6 Sub-Alpha-Complex in High Dimension

Marc Glisse (INRIA – Orsay, FR), Jeff M. Phillips (University of Utah – Salt Lake City, US)

License  Creative Commons BY 4.0 International license
© Marc Glisse, Jeff M. Phillips

5.6.1 Problem Statement

Goal. compute (persistent) homology of low dimension of a union of balls in high dimension.

Usual approaches.

1. Čech (or Rips) of size $n^{(k+1)}$.
2. Alpha-complex which “requires” computing the full Delaunay triangulation of size (up to) $n^{(d/2)}$. This is only better than Approach 1 for small d .
3. Handle each simplex using linear/quadratic programming [1].

Question. Can we improve on it?

- we don’t specifically need the alpha-complex, something closer to the wrap complex would be even better
- more output sensitive?
- use more LP before QP?
- etc

5.6.2 Progress

It was also suggested to maybe we use function approximation coresets to get some sort of approximation relevant for persistence (diagram) approximation using function approximation coresets. For instance in the line of work by Langberg and Schulmann [2] or and more recently (and perhaps more generally) by Alishahi and Phillips [3]. Constructions like Growing Neural Gas [4] could be viewed as an approximation of what we want, and could be further generalized. Funke and Ramos work on smooth-surface reconstruction [5] might also be helpful.

References

- 1 Erik Gunnar Carlsson and John Carlsson. Computing the alpha complex using dual active set quadratic programming. arXiv report, 2023. URL: <https://arxiv.org/abs/2310.00536>.
- 2 Michael Langberg and Leonard J. Schulman. Universal epsilon-approximators for integrals. In Moses Charikar, editor, *21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 598–607, 2010. doi:10.1137/1.9781611973075.50.
- 3 Meysam Alishahi and Jeff M. Phillips. No dimensional sampling coresets for classification. In *41st International Conference on Machine Learning (ICML)*. OpenReview.net, 2024. URL: <https://openreview.net/forum?id=jS3CMHtYJD>.
- 4 Bernd Fritzsche. A growing neural gas network learns topologies. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 625–632. MIT Press, 1994. URL: https://proceedings.neurips.cc/paper_files/paper/1994/hash/d56b9fc4b0f1be8871f5e1c40c0067e7-Abstract.html.
- 5 Stefan Funke and Edgar A. Ramos. Smooth-surface reconstruction in near-linear time. In David Eppstein, editor, *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 781–790, 2002. URL: <http://dl.acm.org/citation.cfm?id=545381.545484>.

5.7 Treewidth of 3-Manifolds

Benjamin Burton (University of Queensland – Brisbane, AU), Kristóf Huszár (TU Graz, AT), Jonathan Spreer (University of Sydney, AU), Yelena Yuditsky (ULB-Brussels, BE)

License © Creative Commons BY 4.0 International license
© Benjamin Burton, Kristóf Huszár, Jonathan Spreer, Yelena Yuditsky

5.7.1 Problem Statement

The *treewidth* $\text{tw}(M)$ of a compact 3-manifold M is defined as the smallest treewidth the dual graph $\Gamma(T)$ of any generalized triangulation T of M may have. In other words

$$\text{tw}(M) = \min\{\text{tw}(\Gamma(T)) : T \text{ is a triangulation of } M\}.$$

The study of $\text{tw}(M)$ is motivated by existing fixed-parameter tractable (FPT) algorithms to efficiently compute various 3-manifold invariants from triangulations with dual graph of bounded treewidth [1, 2, 3, 4, 5] – several of which are known to be NP-hard in general. The treewidth has been related to various other properties of 3-manifolds, such as Heegaard genus [7, 9], hyperbolic volume [10] (cf. [6]), or JSJ decompositions [8], but several fundamental questions remain to be answered.

Question 1. What is the complexity of computing the treewidth of a 3-manifold?

It is natural to ask, what is the computational complexity of the treewidth of a 3-manifold given by a triangulation. We believe that it is at least as hard as computing the treewidth of a graph, which is known to be NP-hard. We know that any algorithm to compute the treewidth of a 3-manifold yields (via a polynomial-time reduction) a constant-factor approximation algorithm for the treewidth of bounded-degree graphs, see [8, Remark 15].

Question 2. What is the treewidth of the 3-torus?

The 3-torus $\mathbb{T}^3 = \mathbb{S} \times \mathbb{S} \times \mathbb{S}$ is defined as the Cartesian product of three circles. Perhaps surprisingly, the exact value of $\text{tw}(\mathbb{T}^3)$ is unknown. It is known that it is at least two (since

3-manifolds with treewidth at most one have been classified [7, Theorem 2]) and at most four (as shown by its minimal triangulation, which has treewidth four [12, Closed Census]).

Question 3. Can we classify closed 3-manifolds of treewidth two?

Given the classification of closed 3-manifolds with treewidth one, it is natural to ask for a similar result regarding 3-manifolds with treewidth two. We know that all Seifert fibered spaces over the 2-sphere or a non-orientable surface have treewidth of at most two [7, Section 5]. Answering Question 3 would also help us get closer to answering Question 2.

Question 4. What is the relationship between minimal triangulations of 3-manifolds and triangulations with minimal treewidth?

We know that minimal triangulations of 3-manifolds (as measured by the number of tetrahedra) are not always of minimal treewidth: the minimal triangulation of the *Poincaré homology sphere* consists of five tetrahedra and its dual graph is K_5 , whose treewidth is four. At the same time, the Poincaré homology sphere is a Seifert fibered space over the 2-sphere, and as such it has a (larger) triangulation with a dual graph of treewidth two [7, Corollary 20]. This is the only example of this kind that we know of.

Question 5. Is the HOMEOMORPHISM PROBLEM of triangulated 3-manifolds FPT in the parameter $k = \max\{\text{tw}(\Gamma(T_1)), \text{tw}(\Gamma(T_2))\}$?

The HOMEOMORPHISM PROBLEM is a fundamental decision problem in computational topology. It asks whether two given triangulated d -manifolds T_1 and T_2 are homeomorphic. For $d = 2$, the problem can be solved in polynomial time, but for $d \geq 4$ it is undecidable. In dimension $d = 3$ it is known to be at least as hard as the GRAPH ISOMORPHISM PROBLEM, but is elementary recursive. See [11, Section 3.1] for context and details.

5.7.2 Progress

During the seminar, we focused primarily on Questions 2 and 3.

Question 2. With the help of *Regina* [12], we ran through all 9,944,398 triangulations of the 3-torus that can be accessed from its minimal triangulation using Pachner moves (bistellar flips) never exceeding 14 tetrahedra at any step. For each triangulation, we used a greedy algorithm to compute a small-width tree decomposition of its dual graph. This greedy algorithm *probably* finds the real treewidth for such small graphs, though this is not guaranteed. We did not find anything with width less than 4 in this list. The following code was used to perform this task in *regina-python*.

```
def test(sig, tri):
    print(TreeDecomposition(tri).width(), sig)
    return False

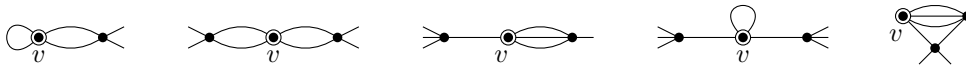
Example3.threeTorus().retriangulate(8, 1, test)
```

The first parameter 8 above means at most eight extra tetrahedra at any stage (so at most 14 in total, since the minimal triangulation has 6 tetrahedra). The second parameter 1 instructs the program to run in single-thread mode.

Question 3. In regard to the classification of closed 3-manifolds with treewidth two the following result may serve as a starting point.

► **Theorem 1** (folklore; adapted from [13, Theorem 3(i)]). *If a multigraph $G = (V, E)$ has treewidth at most two and is not the empty graph, then G has a vertex of degree at most two. (Here the degree of a vertex $v \in V$ denotes the number of other vertices in V adjacent to v .)*

Based on Theorem 1 we started exploring the following approach. Given a triangulation T of a closed 3-manifold with $\text{tw}(\Gamma(T)) = 2$, pick a vertex v in $\Gamma(T)$ of degree at most two. Observe that, for the 1-neighborhood of v in $\Gamma(T)$ there are five possibilities (Figure 1). Each of these five cases corresponds to a finite number of admissible substructures within a valid triangulation of a 3-manifold. Using **Regina**, we enumerated these substructures. Aiming for a recursive classification method, our next goal is to understand how the topology of the underlying manifold changes when we “reduce” the triangulation T around the tetrahedron corresponding to the vertex v .



■ **Figure 1** The five possible local pictures around a vertex $v \in \Gamma(T)$ of degree at most two.



References

- 1 Benjamin A. Burton and Rodney G. Downey. Courcelle’s theorem for triangulations. *J. Comb. Theory A*, 146:264–294, 2017. doi:10.1016/J.JCTA.2016.10.001.
- 2 Benjamin A. Burton, Thomas Lewiner, João Paixão, and Jonathan Spreer. Parameterized complexity of discrete Morse theory. *ACM Trans. Math. Softw.*, 42(1):6:1–6:24, 2016. doi:10.1145/2738034.
- 3 Benjamin A. Burton, Clément Maria, and Jonathan Spreer. Algorithms and complexity for Turaev–Viro invariants. *J. Appl. Comput. Topol.*, 2(1-2):33–53, 2018. doi:10.1007/S41468-018-0016-2.
- 4 Benjamin A. Burton and William Pettersson. Fixed parameter tractable algorithms in combinatorial topology. In Zhipeng Cai, Alex Zelikovskiy, and Anu G. Bourgeois, editors, *20th International Conference Computing and Combinatorics (COCOON)*, volume 8591 of *LNCS*, pages 300–311. Springer, 2014. doi:10.1007/978-3-319-08783-2_26.
- 5 Benjamin A. Burton and Jonathan Spreer. The complexity of detecting taut angle structures on triangulations. In Sanjeev Khanna, editor, *Proceedings 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 168–183, 2013. doi:10.1137/1.9781611973105.13.
- 6 Kristóf Huszár. On the pathwidth of hyperbolic 3-manifolds. *Comput. Geom. Topol.*, 1(1):1:1–1:19, 2022. URL: <https://www.cgt-journal.org/index.php/cgt/article/view/4>.
- 7 Kristóf Huszár and Jonathan Spreer. 3-manifold triangulations with small treewidth. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry (SoCG)*, volume 129 of *LIPICs*, pages 44:1–44:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.SOCG.2019.44.
- 8 Kristóf Huszár and Jonathan Spreer. On the width of complicated JSJ decompositions. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG)*, volume 258 of *LIPICs*, pages 42:1–42:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.SOCG.2023.42.
- 9 Kristóf Huszár, Jonathan Spreer, and Uli Wagner. On the treewidth of triangulated 3-manifolds. *J. Comput. Geom.*, 10(2):70–98, 2019. URL: <https://doi.org/10.20382/jogc.v10i2a5>, doi:10.20382/JOGC.V10I2A5.
- 10 Clément Maria and Jessica S. Purcell. Treewidth, crushing and hyperbolic volume. *Algebr. Geom. Topol.*, 19(5):2625–2652, 2019. doi:10.2140/agt.2019.19.2625.

- 11 Marc Lackenby. Algorithms in 3-manifold theory. In *Surveys in differential geometry 2020. Surveys in 3-manifold topology and geometry*, volume 25 of *Surv. Differ. Geom.*, pages 163–213. Int. Press, Boston, MA, 2022.
- 12 Benjamin A. Burton, Ryan Budney, William Pettersson, et al. Regina: Software for low-dimensional topology. <http://regina-normal.github.io/>, 1999–2023.
- 13 Hans L. Bodlaender, Benjamin A. Burton, Fedor V. Fomin, and Alexander Grigoriev. Knot diagrams of treewidth two. In Isolde Adler and Haiko Müller, editors, *46th International Workshop Graph-Theoretic Concepts in Computer Science (WG)*, volume 12301 of *LNCS*, pages 80–91. Springer, 2020. doi:10.1007/978-3-030-60440-0_7.

5.8 WSPD for Low Density Graphs

Joachim Gudmundsson (*The University of Sydney, AU*)

License  Creative Commons BY 4.0 International license
 Joachim Gudmundsson

5.8.1 Problem Statement

Given a geometric graph $G = (V, E)$ where the vertices are points in the plane and the edge weights is the Euclidean distance between its endpoints, the main goal is to construct a $(1 + \varepsilon)$ -approximate distance oracle ($O(1)$ query time) using linear space and $O(m + n \log n)$ preprocessing.

Of course this is not possible for arbitrary geometric graphs, instead we want to investigate restricted graph classes. In particular, we want to investigate graph classes that include road networks (think about queries in Google maps). Such a graph class should allow for edge intersections and include naturally occurring graphs such as grid graphs or “cul-de-sac” graphs. There are a number of “realistic” graph classes, e.g. planar, low-dilation, c -packed, small highway dimension, small skeleton dimension, and small doubling dimensions that allow for approximate distance oracles, but these graph classes do not include road networks.

There are a few graph classes that include road networks, e.g. disk neighbourhood, crossing degeneracy, and bounded growth graphs but they do not have efficient algorithmic tool sets. Our favourite natural graph class that includes road networks is λ -low density graphs or the more general class of τ -lanky graphs. A geometric graph is λ -low density if for every p and r the ball $B(p, r)$ with centre at p and radius r intersects at most λ edges of E of length at least r .

In a recent paper [1] we showed that $O(1)$ -low-density graphs have a $(1 + \varepsilon)$ -approximate distance oracle using $O(n \log n)$ space and $O(n^{3/2} \text{polylog} n)$ preprocessing. The key insight used to obtain this result is a Well-Separated Pair Decomposition (WSPD) of V in the graph metric of $O(n \log n)$ size (which gives the space bound). There are two natural techniques that could be used to improve this result:

1. Improve the size of the WSPDs for low density graphs. No non-trivial lower bound is known. Can we close the gap between $\Omega(n)$ and $O(n \log n)$? Unit disk graphs (UDG) have the same bounds. Even subgraphs of a grid graph would be interesting (or even a tree in a grid graph). In dimensions greater than 2 the lower bound for the WSPD for UDG is $\Omega(n^{2-2/d})$.
2. Find a tree cover T_1, \dots, T_k of G of constant size and $(1 + \varepsilon)$ dilation. A tree cover is a set of trees s.t. for every pair of points p and q (a) there exists a tree T_i with $d_{T_i}(p, q) \leq (1 + \varepsilon) \cdot d_G(p, q)$ and (b) for every tree T_i $d_{T_i}(p, q) \geq d_G(p, q)$. If such a tree cover exists we would probably get a $(1 + \varepsilon)$ -approximate distance oracle of linear space.

Any results for 1 or 2 would give improved results.

References

- 1 Joachim Gudmundsson and Sampson Wong. A well-separated pair decomposition for low density graphs. *CoRR*, abs/2411.08204, 2024. doi:10.48550/ARXIV.2411.08204.

5.9 Consistent System of Shortest Paths

Francis Lazarus (CNRS, Grenoble, FR), Jeff Erickson (University of Illinois – Urbana-Champaign, US), Raimund Seidel (Universität des Saarlandes – Saarbrücken, DE), and Ling Zhou (Duke University, Durham, US)

License © Creative Commons BY 4.0 International license
© Francis Lazarus, Jeff Erickson, Raimund Seidel, Ling Zhou

This problem has been resolved during the seminar. More precisely, the problem was already solved by several authors, starting with Hartvigsen and Mardon [6].

5.9.1 Problem Statement

Given a graph $G = (V, E)$ with a weight function $w : E \mapsto \mathbb{R}_+$, the problem is to compute a *consistent* system \mathcal{P} of shortest paths w.r.t. to w , so that for every $u, v \in V$ there is a shortest path from u to v in \mathcal{P} and for every paths $p, q \in \mathcal{P}$ their intersection is connected (possibly empty), i.e. is a common subpath.

If shortest paths are unique for the weight function, then the set of shortest paths is consistent. There are various perturbation schemes that ensure uniqueness of shortest paths, either non-deterministic or deterministic but with an extra overhead in complexity (see discussion below). The goal here is to produce a consistent system of shortest paths in a deterministic way with no overhead cost compared to the traditional all-pair-shortest-paths algorithms. The hope is to compute a consistent system in a more efficient way, without enforcing uniqueness of shortest paths...

Computing a consistent system of shortest paths appears to be equivalent to compute a *consistent system of rooted shortest path trees*. Here, consistent means that for every $u, v \in V$, the (shortest) path from u to v in the tree T_v rooted at v is the reverse of the path from v to u in the tree T_u rooted at u , i.e. $T_v(u) = (T_u(v))^{-1}$

This last characterization is itself equivalent to the following: for every $u, v \in V$

1. $T_v(u)$ and $T_u(v)$ have the same number of edges, and
2. denoting by π_u the parent pointer in T_u , either $\pi_v(u) = v$ or $\pi_v(u) = \pi_{\pi_u(v)}(u)$.

The first condition is easily enforced by adding a constant perturbation to the edge weights. The second is “local” in the sense that it only involves pointer comparisons (as opposed to compare the paths $T_v(u)$ and $T_u(v)$).

5.9.2 Progress

A simple randomized perturbation of the edge weights provides uniqueness of shortest paths with high probability. The perturbations are obtained by drawing independently and uniformly at random an integer in $[1, n^4]$ for every of the n edges. See [1, Sec 6.1] for a full description of this non-deterministic perturbation scheme based on the Isolating Lemma of Mulmey, Vazirani and Vazirani [2].

There are two known deterministic approaches to compute weight perturbations to guarantee uniqueness of shortest paths for graphs embedded on surfaces.

- One uses lexicographic perturbation, incurring a $O(\log n)$ overhead [1]. This perturbation scheme assigns a unique index to each edge in the graph, and then breaks ties between shortest paths by comparing the lexicographically smallest index where the two paths differ. The scheme uses a dynamic-forest data structure to maintain a shortest-path tree during its construction via Dijkstra’s algorithm. In addition to uniqueness, this scheme guarantees that shortest paths are symmetric: The reversal of any shortest path is also a shortest path.
- The other scheme uses homology and incurs a $O(g)$ -factor overhead [3]. This “holiest” perturbation scheme uses a vector of $2g + 1$ perturbation terms for each edge, $2g$ of which are homology signatures, and the last of which stores the number of vertices in a fixed dual spanning tree on one side of the edge. Path lengths are vectors of length $2g + 2$, which are added as vectors and compared lexicographically. This is a generalization of the leftmost-shortest path tie-breaking rule for planar graphs, originally proposed by Ford and Fulkerson. Like the leftmost shortest paths in planar graphs, holiest shortest paths are not necessarily symmetric. Consider two opposite corner vertices in a square unweighted grid graph.

It should be emphasized that the main problem here is not making individual edge weights distinct, but making certain sums of edge weights distinct.

Raimund Seidel points out that the Floyd-Warshall all-pairs shortest path algorithm computes a consistent set of shortest paths. Recall that the i th iteration of Floyd-Warshall computes the shortest path from each vertex u to each vertex v whose intermediate vertices have index at most i . It follows that FW computes shortest paths that lexicographically minimize the sorted vector of vertex indices.

Unfortunately, Floyd-Warshall runs in $O(n^3)$ time, which is more than we want to spend for most surface-graph algorithms.

It appears that Wulff-Nilsen already described a similar perturbation scheme for arbitrary graphs that allows Dijkstra to compute paths that minimize length, then the number of edges, and finally the sorted vector of vertex indices, all in $O(n \log n)$ time [4] for a single shortest path tree. By repeating Dijkstra from every vertex we thus obtain a consistent system of shortest paths in $O(n^2 \log n)$ time. Note that Wulff-Nilsen uses $O(\log n)$ extra pointers per vertex, so that running a single (modified) Dijkstra actually uses $O(n \log n)$ space instead of the usual linear space implementation. In fact, Hartvigsen and Mardon [6] formerly described an implementation that runs all the Dijkstra’s in parallel in $O(n^2 \log n)$ time without using extra pointers.

See also Borradaile, Sankowski, and Wulff-Nilson [5, Sec. 7] for an implementation using top trees.

This is the basis for the $O(\log n)$ -factor lexicographic perturbation scheme described above; the $O(\log n)$ overhead does not apply to Dijkstra’s algorithm, but rather to the pivots inside our multiple-source shortest paths algorithm. Similarly, Borradaile, Sankowski, and Wulff-Nilson show that more modern shortest-path algorithms based on dense-distance graphs incur a $O(\log^2 n)$ overhead.

References

- 1 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-source shortest paths in embedded graphs. *SIAM Journal on Computing*, 42(4):1542–1571, 2013. <https://arxiv.org/pdf/1202.0314>.

- 2 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
- 3 Jeff Erickson, Kyle Fox, and Luvsandondov Lkhamsuren. Holiest minimum-cost paths and flows in surface graphs. In *50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2018. URL: <https://arxiv.org/pdf/1804.01045>, doi:10.1145/3188745.3188904.
- 4 Christian Wulff-Nilsen. Minimum cycle basis and all-pairs min cut of a planar graph in subquadratic time. arXiv report, 2009. doi:10.48550/arXiv.0912.1208.
- 5 Glencora Borradaile, Piotr Sankowski, and Christian Wulff-Nilsen. Min st-cut oracle for planar graphs with near-linear preprocessing time. *ACM Transactions on Algorithms*, 11(3):1–29, 2015. doi:10.1145/2684068.
- 6 David Hartvigsen and Russell Mardon. The all-pairs min cut problem and the minimum cycle basis problem on planar graphs. *SIAM Journal on Discrete Mathematics*, 7(3):403–418, May 1994. doi:10.1137/s0895480190177042.

5.10 Graph Tiles

Oswin Aichholzer (TU Graz, AT), Robert Ganian (TU Wien, AT), Phillip Keldenich (TU Braunschweig, DE), Maarten Löffler (Utrecht University, NL), Gert Meijer (NHL Stenden Hogeschool – Leeuwarden, NL), Alexandra Weinberger (FernUniversität in Hagen, DE), Carola Wenk (Tulane University – New Orleans, US)

License © Creative Commons BY 4.0 International license

© Oswin Aichholzer, Robert Ganian, Phillip Keldenich, Maarten Löffler, Gert Meijer, Alexandra Weinberger, Carola Wenk

5.10.1 Problem Statement

In a *graph tiling* problem, we are given as input a set of square tiles with pieces of graph drawn (embedded) on them, and the output is an arrangement of those tiles into a larger area, like in Figure 2.

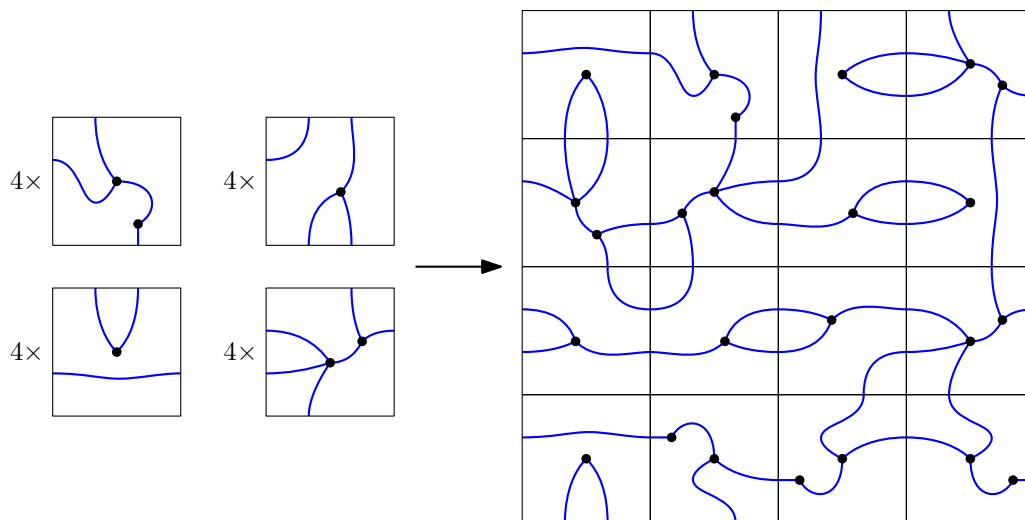
Now, we are interested in output arrangements that satisfy certain properties.

- Tiles should “match up” on the sides.
- We may want that the resulting graph is connected.
- We may want to minimize the length of longest path between any two vertices in the resulting graph.
- We may want to minimize the area of the largest face, or maximize the area of the smallest face.

In general, even satisfying the first property alone is already NP-hard. However, this is not the case when there is only a single type of tile. Can we parametrize this problem by the number and/or complexity of the tiles? And for really simple sets of tiles, how many of these properties can we still test for efficiently?

The following variants and limitations might also be interesting to consider, and may influence to complexity of the problem.

- One can consider the tiles to be arranged into different shapes.
 - A square
 - A rectangle
 - A polygon without holes
 - A polygon with holes
 - ...



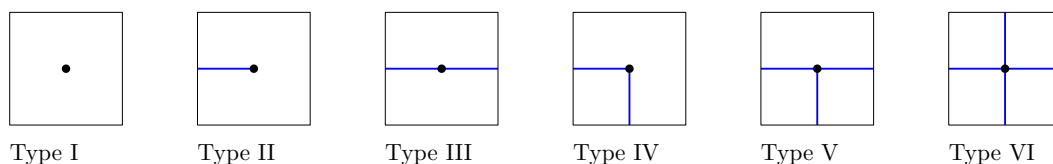
■ **Figure 2** A graph tiling.

- For the boundary of the form (where tiles do not meet other tiles and so are not restricted by having to match another tile) we can ...
 - allow everything.
 - prescribe something (natural would be that all are empty or all have a path going in or there is a prescription for each).
 - avoid having a boundary by putting the tiles on a torus.
 - ...
- Restrictions on the graph other than (or in addition to) connectedness could also be made. For example, one could require the graph to be a tree.

5.10.2 Progress

During the seminar, we considered the following setting: Given n^2 tiles that have constantly many different types. Can all the tiles be arranged into one square such that the sides of the tiles match and the boundary is arbitrary?

When we restrict the tile types to the six types of tiles from Figure 3 and allow rotation, then if all tiles are of the same type, it is always possible to arrange them into a valid square. If tiles are of exactly two types, then we have polynomial time algorithms to decide if they can be arranged into a square. (For most combinations, the answer is always yes or always no.) For three types most cases still give easy polynomial time algorithms, but some cases are still open.



■ **Figure 3** Six types of tiles.

5.11 Approximate Minlink Paths in Subquadratic Time

Valentin Polishchuk (*Linköping University, SE*), Sayan Bandyapadhyay (*Portland State University, US*), Mark de Berg (*TU Eindhoven, NL*), Joachim Gudmundsson (*The University of Sydney, AU*), André Nusser (*CNRS, FR*), Frank Staals (*Utrecht University, NL*)

License © Creative Commons BY 4.0 International license

© Valentin Polishchuk, Sayan Bandyapadhyay, Mark de Berg, Joachim Gudmundsson, André Nusser, Frank Staals

5.11.1 Problem Statement

Let P be a polygonal domain with n vertices. A minlink path between points s, t in P is an s - t path with fewest edges (links). Computing the path is 3SUM-hard, and an $O(\sqrt{n})$ approximation can be found in subquadratic time [1]. Can a better approximation be computed in subquadratic time?

5.11.2 Progress

The following are sketches of ideas that arose during the seminar.

$O(n \log n)$ space, $O(\log n)$ query, +2 approx

Build a balanced hierarchical subdivision. For the bottom endpoint; construct a SPM with respect to the link distance (which uses $O(n)$ space). Recurse.

Space: $O(n \log n)$

Query: For $LCA(s, t)$, and all its ancestors v_1, \dots, v_k , query the min-link path from s to v_i and from v_i to t . Report the cheapest path. Beschreibung

► **Claim 1.** This is a +2 approx.

Proof. Look at min-link path from s to t : If s, t are on opposite sides of chord through v_i , the approx ratio is +2. If s, t are on same side of v_i , the approx is +1. ◀

$O(n^2)$ space, $O(\log n)$ query

Let v be the first vertex on the geodesic shortest path from s to t .

► **Claim 2.** There exists a min-link path from s to t that passes through v .

Consider the (geodesic) SPM of v , and consider the min-link diagram $LPM(v, \phi)v$, in which the first link has fixed orientation ϕ .

► **Claim 3.** Any edge of $LPM(v, \phi)$ is contained in a triangle of $SPM(v)$.

Proof sketch. Assume by contradiction that $e = wq$ of $SPM(v)$ intersects an edge of $LPM(v, \phi)$ in some point p . Now consider two points p_1 and p_2 on e on opposite sides of p . This means the min-link distance from v (with orientation ρ) to p_1 is i , and the min-link distance from v to p_2 is $j \neq i$.

However, by Claim 2 there exists a min-link path from p_1 to v or s that goes through w . Similarly, there exists a min-link path from p_2 to v or s that goes through w . They must have the same length. Contradiction. ◀

(We may want to argue that an edge of $LMD(s)$ cannot intersect an edge of $SPM(v)$.)
So, for any triangle of $SPM(v)$, there is at most one edge $e(\phi)$ of $LPM(v, \phi)$ in it.

► **Claim 4.** $e(\phi)$ rotates monotonically as a function of ϕ (or it is fixed).

So, for any triangle; we store the interval $(-\infty, \phi_1]$ on which the triangle is simply contained in a “length $i + 1$ ” LPM region, and the interval $[\phi_2, \infty)$ in which it is “length $i + 1$ ”. In the interval $[\phi_1, \phi_2]$ we store the function describing how $e(\phi)$ moves.

► **Claim 5.** For any point q , the link-distance from v to q whose first link has orientation ϕ is either i or $i + 1$ (over all ϕ s.)

Hence, for every vertex v we have some $O(n)$ space structure. (The SPM, and per triangle the description of one $LPM(v, \phi_i)$ edge).

So $O(n^2)$ in total. In addition we store the Guibas Hershberger structure. Space remains $O(n^2)$.

Query: We locate v using the Guibas Hershberger structure. This also gives us the orientation ϕ . We now point locate t in the $SPM(v)$; we then compare ϕ against the interval $[\phi_{i1}, \phi_{i2}]$ and report either i or $i + 1$.

$O(n)$ space $O(k \log n)$ query

Main idea. We simulate the min-link propagation using the Guibas Hershberger structure.

Store the Guibas Hershberger structure: Compute the first vertex v on geodesic shortest path from s to t . There exists a min-link path through v .

Consider the window $w_1 = vq$ on the ray from s through v . Consider the weak visibility polygon V of w_1 . We essentially want to compute which pocket of $P \setminus V$ contains t . Let w_2 be the window (of V) separating s from t .

Query the funnel from t to v and q (that has apex a).

► **Claim 6.** a is a vertex of w_2 .

Main idea. We can use the Guibas Hershberger structure to actually find a in $O(\log n)$ time. We can then use a ray shooting query to compute the other endpoint of w_2 (i.e. by extending the “middle” triangle of the funnel in the other direction”). So, we simulate one step of the visibility propagation in $O(\log n)$ time; this gives us $O(k \log n)$ query time.

$O(n^2)$ space, $O(\log n)$ query, +4 approx

For every pair of vertices, store the minimum link distance.

Store a triangulation of the free space.

Locate the triangles containing s and t . Let v and w be vertices incident to these triangles. Report 2+ length min-link path (v, w) .

► **Claim 7.** This is a +4 approx for $d(s, t)$.

$O(n^4)$ space, $O(\log^c n)$ query, +2 approx

Rather than picking some arbitrary first vertex v and last vertex w , we actually find the best pair.

Let v, w be the pair of vertices (v visible from s , w visible from t) for which min-link distance from v to w is minimum. We will report the path s to v to $\dots w$ to t .

► **Claim 8.** This is a +2 approx.

Main idea. We build some three-level structure to efficiently find v and w .

For every vertex, compute its visibility polygon, store it in a DS for stabbing queries, so that we can report the polygons stabbed by s as $O(\log n)$ canonical subsets. For each such canonical subsets we again store a structure for stabbing queries with t as $\log n$ canonical subsets. As a third level structure: simply store the link distance.

(This is very similar to the $2 - pt$ geodesic query structure Sarita de Berg, Tillmann Miltzow, and Frank Staals [2] constructed before.)

$O(n^8)$ space, $O(\log n)$ query, +1 approx

The vague idea is the following: We do the same as above, except that for every vertex v , we compute its minimum link map (which has $O(n^4)$ complexity or so); we throw all these regions into our stabbing structure. (This may also require us to pick the right angle from which we leave v .)

$O(n^{1+1/k})$ space, $O(\log n)$ query, *2 approx

Main idea. Same as our $O(n^2)$ space structure, except that we store approximate distances between every pair of vertices.

There exists an $O(n^{1+1/k})$ space structure to store distances in an arbitrary graph [3]. We use this on the complete graph whose edge lengths are the link-distances.

References

- 1 Joseph S. B. Mitchell, Valentin Polishchuk, and Mikko Sysikaski. Minimum-link paths revisited. *Comput. Geom.*, 47(6):651–667, 2014. URL: <https://doi.org/10.1016/j.comgeo.2013.12.005>, doi:10.1016/J.COMGEO.2013.12.005.
- 2 Sarita de Berg, Tillmann Miltzow, and Frank Staals. Towards space efficient two-point shortest path queries in a polygonal domain. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPICs*, pages 17:1–17:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.SOCG.2024.17.
- 3 Shiri Chechik. Approximate distance oracles with constant query time. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 654–663. ACM, 2014. doi:10.1145/2591796.2591801.

5.12 Alexander Theorem for Knotted Spheres

Mathijs Wintraecken (Centre Inria d’Université Côte d’Azur, FR) and Kristóf Huszár (TU Graz, AT)

License © Creative Commons BY 4.0 International license
© Mathijs Wintraecken and Kristóf Huszár

5.12.1 Problem Statement

Alexander [1] proved that every knot or even link can be represented as a closed braid, that is, a braid with ends identified. This result is known as the Alexander theorem. The question is if this has a generalisation to knotted spheres in the following sense: Can every knotted 2-sphere in \mathbb{R}^4 be embedded sufficiently close to the standard symmetrically embedded 2-sphere $\mathbb{S}^2 \subset \mathbb{R}^3 \subset \mathbb{R}^4$ such that the closest point projection onto the symmetrically embedded 2-sphere is a local diffeomorphism? Has this question been answered before?

5.12.2 Progress

With the help of ChatGPT, Kristóf Huszár found the appropriate literature references, of which [2, Chapter 1] already contains this result with a pedagogical explanation.

References

- 1 James W. Alexander. A lemma on systems of knotted curves. *Proceedings of the National Academy of Sciences*, 9(3):93–95, 1923. doi:10.1073/pnas.9.3.93.
- 2 Scott Carter, Seiichi Kamada, and Masahico Saito. *Surfaces in 4-Space*. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-662-10162-9.

5.13 Universal Point Sets

Alexander Wolff (Universität Würzburg, DE), Stefan Felsner (TU Berlin, DE), Michael Hoffmann (ETH Zürich, CH), Maarten Löffler (Utrecht University, NL), Fabrizio Montecchiani (University of Perugia, IT), Yoshio Okamoto (The University of Electro-Communications, JP), Jonathan Spreer (University of Sydney, AU)

License  Creative Commons BY 4.0 International license

© Alexander Wolff, Stefan Felsner, Michael Hoffmann, Maarten Löffler, Fabrizio Montecchiani, Yoshio Okamoto, Jonathan Spreer

5.13.1 Problem Statement

A set S of points in the plane is called *universal* with respect to a given drawing style and an integer n if every graph on n vertices admits a drawing in the given style such that the vertices are mapped to the points of S . For planar graphs, for example, the drawing style could require crossing-free straight-line edges. When considering drawings where the edges are circular arcs [1] or polygonal lines [4], the encoding of the edges may also need to be considered; for instance, a third point on an arc or the bends of a polygonal line. Also the number of edges can be a relevant parameter, which leads to the notion of *universal geometric (or topological) graphs* [2]. Another direction is to study universal point sets for restricted classes of planar graphs, such as planar 3-trees [3].

5.13.2 Progress

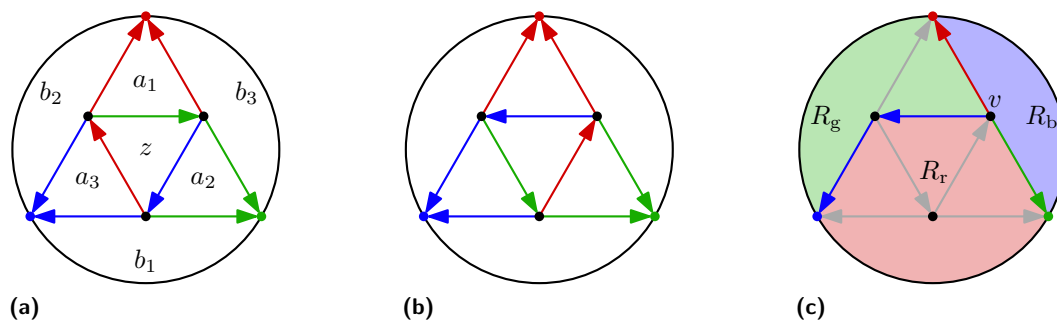
Higher dimensions. One possible way to transfer the problem of universal point sets into higher dimensions is the following: Can we find a universal point set in 3-space for all n -vertex triangulations of the 2-sphere?

By a classical result due to Steinitz, every triangulated 2-sphere can be embedded into 3-space in convex position (in particular, as a polyhedral embedding – that is, with straight-line edges and flat triangles). Moreover, we know that all of these embeddings can be chosen to have integer coordinates.

By a result of André Schulz [5], the maximum coordinate of such an integer embedding is $O(2^{7.55n})$, where n is the number of vertices. This trivially yields a universal point set of size $(c \cdot 2^{7.55n})^3$. This result holds for arbitrary polytopes. The hope is that the bound for simplicial polytopes (triangulations of the 2-sphere) would be much smaller. There might be hints in the proof of the above result that can help us.

This touches on a famous problem in discrete geometry: The polyhedral realization problem for triangulated (or polyhedral) surfaces. Not every triangulation of an orientable surface has a polyhedral embedding into 3-space. In fact, an n -vertex triangulation of an orientable surface can have genus quadratic in n , but the highest genus surfaces that are known to be polyhedrally embeddable have genus $n \log(n)$. Even more, deciding polyhedral embeddability for a given triangulation is likely a hard problem in general. See [6] for a nice overview of the polyhedral realization problem.

Central vertices in Schnyder woods. To construct universal point sets in the plane, it seems natural to find a central vertex in the given planar graph, map it to a central point of the point set, split the graph into, say, three regions that meet in the central vertex, split the point set accordingly, and recurse. In fact, in a Schnyder wood of a 3-connected plane graph G , every vertex does have three associated regions, which form a partition of the inner faces of G ; see the regions R_r , R_g , and R_b of vertex v in Figure 4c. A vertex is δ -central if each of its regions contains at most $(1 - \delta)f_0$ faces of G , where f_0 is the number of inner faces of G . The centrality of the outer vertices is 0. Here we show that every Schnyder wood has a $1/3$ -central vertex.



■ **Figure 4** The two Schnyder woods of the octahedron (a,b) and the three regions of a vertex v (c).

► **Proposition 1.** If S is a Schnyder wood of a plane 3-connected graph G with f_0 bounded faces, then there is a vertex c that is $1/3$ -central with respect to S .

The proof, which we omit here, uses Sperner’s Lemma (see “Proofs from the book” by Aigner and Ziegler). Next we show that the above centrality bound is tight.

► **Proposition 2.** For every $\varepsilon > 0$, there exists a planar triangulation T with f_0 bounded faces such that, in every Schnyder wood of T , no vertex has centrality greater than $(1/3 + \varepsilon)$.

Proof. Consider the skeleton of the octahedron; see Figure 4. We first consider a weighted version of the partition problem where we assign a weight w_f to each face f among the seven inner faces; namely $w(z) = w(b_1) = w(b_2) = w(b_3) = 1$ and $w(a_1) = w(a_2) = w(a_3) = \alpha$ for some real $\alpha > 0$. The total weight of the faces is $3\alpha + 4$. Each inner vertex has a region of weight $2\alpha + 2$. Hence, with increasing α , the weight of a region of each vertex can get arbitrarily close to $2/3$ of the total weight of the faces of the graph.

We can subdivide the faces of the octahedron and use the weights to count the number of (sub)faces in each original octahedron face. To this end, we place a triangulation with k inner vertices into each of the faces a_1 , a_2 , and a_3 , and set $\alpha = 2k + 5$. In every Schnyder wood of the new triangulation with $3k + 6$ vertices, the edges of the original octahedron are still colored and oriented in one of the two ways shown in Figures 4a and 4b. Let $i \in \{1, 2, 3\}$. If u is a vertex of the triangulation that refines the octahedron face a_i , then the largest region of u contains z , b_i , and all the faces in the triangulations subdividing a_{i-1} and a_{i+1} . Hence u has a region whose weight exceeds $2\alpha + 2$. ◀

References

- 1 Patrizio Angelini, David Eppstein, Fabrizio Frati, Michael Kaufmann, Silvain Lazard, Tamara Mchedlidze, Monique Teillaud, and Alexander Wolff. Universal point sets for drawing planar graphs with circular arcs. *J. Graph Algorithms Appl.*, 18(3):313–324, 2014. doi:10.7155/jgaa.00324.

- 2 Fabrizio Frati, Michael Hoffmann, and Csaba D. Tóth. Universal geometric graphs. *Combinatorics, Probability and Computing*, 32(5):742–761, 2023. doi:10.1017/S0963548323000135.
- 3 Radoslav Fulek and Csaba D. Tóth. Universal point sets for planar three-trees. *J. Discrete Algorithms*, 30:101–112, 2015. doi:10.1016/J.JDA.2014.12.005.
- 4 Maarten Löffler and Csaba D. Tóth. Linear-size universal point sets for one-bend drawings. In Emilio Di Giacomo and Anna Lubiw, editors, *23rd Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 9411 of *LNCS*, pages 423–429. Springer, 2015. doi:10.1007/978-3-319-27261-0_35.
- 5 André Schulz. *Lifting planar graphs to realize integral 3-polytopes and topics in pseudo-triangulations*. PhD thesis, Freie Universität Berlin, 2008. doi:10.17169/refubium-16284.
- 6 Günter M. Ziegler. *Polyhedral Surfaces of High Genus*, pages 191–213. Birkhäuser Basel, 2008. doi:10.1007/978-3-7643-8621-4_10.

Participants

- Oswin Aichholzer
TU Graz, AT
- Sayan Bandyopadhyay
Portland State University, US
- Benjamin Burton
University of Queensland –
Brisbane, AU
- Otfried Cheong
Scalco, DK
- Mark de Berg
TU Eindhoven, NL
- Jeff Erickson
University of Illinois
Urbana-Champaign, US
- Fedor V. Fomin
University of Bergen, NO
- Robert Ganian
TU Wien, AT
- Joachim Gudmundsson
The University of Sidney, AU
- Dan Halperin
Tel Aviv University, IL
- Michael Hoffmann
ETH Zürich, CH
- Kristóf Huszár
TU Graz, AT
- Phillip Keldenich
TU Braunschweig, DE
- David G. Kirkpatrick
University of British Columbia –
Vancouver, CA
- Francis Lazarus
CNRS – Grenoble, FR
- Jim Little
University of British Columbia –
Vancouver, CA
- Maarten Löffler
Utrecht University, NL
- Gert Meijer
NHL Stenden Hogeschool –
Leeuwarden, NL
- Fabrizio Montecchiani
University of Perugia, IT
- Ian Munro
University of Waterloo, CA
- André Nusser
CNRS – Sophia Antipolis, FR
- Eunjin Oh
POSTECH – Pohang, KR
- Yoshio Okamoto
The University of
Electro-Communications –
Tokyo, JP
- Jeff M. Phillips
University of Utah –
Salt Lake City, US
- Valentin Polishchuk
Linköping University, SE
- Raimund Seidel
Universität des Saarlandes –
Saarbrücken, DE
- Jonathan Shewchuk
University of California –
Berkeley, US
- Diane Souvaine
Tufts University – Medford, US
- Jonathan Spreer
University of Sydney, AU
- Frank Staals
Utrecht University, NL
- Birgit Vogtenhuber
TU Graz, AT
- Hubert Wagner
University of Florida –
Gainesville, US
- Alexandra Weinberger
FernUniversität in Hagen, DE
- Carola Wenk
Tulane University –
New Orleans, US
- Mathijs Wintraecken
Centre Inria d'Université Côte
d'Azur – Sophia Antipolis, FR
- Alexander Wolff
Universität Würzburg, DE
- Yelena Yuditsky
ULB-Brussels, BE
- Ling Zhou
Duke University – Durham, US

