
Geometric problems in higher dimensions:
Voronoi diagrams, the Fermat point, and the
bichromatic discrepancy

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Martin Gerd Suderland

under the supervision of
Prof. Evanthia Papadopoulou

May 2022

Dissertation Committee

Prof. Kai Hormann Università della svizzera italiana, Switzerland
Prof. Piotr Krzysztof Didyk Università della svizzera italiana, Switzerland
Prof. Gill Barequet The Technion—Israel Inst. of Technology, Israel
Prof. Chee Yap New York University, USA

Dissertation accepted on 18 May 2022

Research Advisor
Prof. Evanthia Papadopoulou

PhD Program Director
Prof. Walter Binder

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Martin Gerd Suderland
Lugano, 18 May 2022

Abstract

Research in computational geometry has produced many results in the plane and low dimensional spaces. These include impossibility and complexity results, efficient data structures and optimal algorithms. In higher dimensions the situation is quite different and such results are more sparse. We consider three such problems.

- We study the order- k Voronoi diagram of lines, line segments and convex polyhedra under Euclidean and polyhedral distances. We derive properties and complexity bounds for the diagram and its unbounded features. In case of the farthest Voronoi diagram, we also discuss optimal time algorithms for deriving the unbounded features in the Euclidean version and for constructing the complete diagram when the polyhedral distances are used.
- We propose several algorithms for computing ε -approximations of the Fermat point based on subdivision methods. We put an emphasis on robustness derived through soft predicates and interval arithmetic. We use similar techniques to derive approximations of n -ellipses in the plane. Our implementation and experiments suggest practicability of the proposed methods.
- We introduce a new concept called bichromatic discrepancy. It deals with the question of how uniformly points of two colors can be distributed in the unit square. We derive a lower bound for this bichromatic discrepancy and also describe point sets, which induce a small discrepancy. An application of this new concept to digitalizing Euclidean segments is pointed out.

Acknowledgements

First of all, I would like to thank Evanthia Papadopoulou for supervising my PhD studies. Her advice and guidance was always enormously helpful.

I would also like to thank all members of the dissertation committee - Gill Barequet, Piotr Krzysztof Didyk, Kai Hormann, and Chee Yap - for spending their time to evaluate this thesis.

It was a pleasure to work at the computational geometry group at USI together with my two colleagues Kolja Junginger and Ioannis Mantas. Thank you for both the good work on the common research and also for all the nice free time activities that we did together.

I also thank all my other collaborators with whom I did research related to this thesis. Thank you Gill Barequet for spending your time to get me started with my first paper during my PhD. I am grateful for the opportunity to join Man-Kwun Chiu, Matias Korman, and Takeshi Tokuyama on their project and shortly visiting them in Sendai. Thank you Chee Yap for introducing me to the topic of subdivision methods, in particular during your half-year visit in Lugano. I really enjoyed the Voronoi++ meetings that Franz Aurenhammer organized and which fueled our fruitful joint work.

I am very grateful for all the friends that I got to know in Lugano.

Finally and for most, I want to thank my family, which has been supporting me my entire life.

Contents

Contents	vii
List of Figures	ix
1 Introduction	1
1.1 Computational geometry in high dimensions	1
1.2 Dissertation Overview	2
1.3 List of publications	3
2 Voronoi diagrams of generalized sites	5
2.1 Preliminaries	14
2.1.1 Point-Hyperplane Duality	14
2.1.2 Levels in an Arrangement of Hyperplanes	14
2.1.3 The Gaussian Map	15
2.2 Properties of the Farthest and Order- k Voronoi Diagram	18
2.2.1 Combinatorial Properties	18
2.2.2 Structural Properties	18
2.3 Line Segments as Sites	20
2.4 Lines as Sites	25
2.5 Combination of Lines and Segments	32
2.6 Polyhedra or Clusters of Points as Sites	38
2.7 Conclusion	41
3 On the Trisector of Lines in Three Space	43
3.1 Review: Farthest Voronoi diagram of lines and line segments in \mathbb{R}^2	47
3.2 Towards constructing the FVD of lines and line segments in \mathbb{R}^3 . .	51
4 Farthest Polyhedral Voronoi diagram	53
4.1 Convex polyhedral distance	56
4.2 Farthest-site Voronoi diagram	57

4.3	More properties of FVD	59
4.4	Variants	63
4.5	Conclusion	64
5	Fermat point and n-ellipses	65
5.1	Introduction	65
5.2	Preliminaries	69
5.3	Approximate Fermat points	73
5.3.1	Using the Subdivision Paradigm	74
5.3.2	Enhancing the Subdivision Paradigm	75
5.3.3	Certifying the Weiszfeld method	78
5.4	Approximating n -ellipses	81
5.5	Experiments	85
5.6	Details on box approximations	91
5.6.1	Box approximation of the gradient $\nabla\varphi$	91
5.6.2	Box approximation of φ	92
5.6.3	Box approximation of the Hessian $\nabla^2\varphi$	93
5.7	Conclusion	94
6	Bichromatic discrepancy	97
6.1	Discrepancy lower bound	99
6.2	Point sets with constant discrepancy	104
6.3	Application to digitalizing line segments	109
6.4	Conclusion	112
	Bibliography	115

Figures

2.1	Euclidean nearest Voronoi diagram	6
2.2	Euclidean farthest Voronoi diagram	8
2.3	(left) The Euclidean farthest Voronoi diagram (in grey) of 2 line segments (red and blue). The intensity of the colors encode the proximity of a point to its closest site. Darker colors correspond to smaller distances. (right) The distance functions of both segments in \mathbb{R}^3 . The upper envelope of these distance functions correspond to the diagram on the left.	9
2.4	(left) The Euclidean farthest Voronoi diagram of segments in \mathbb{R}^2 and the intersection with a large circle, which approximates its Gaussian map. Each segment has a different color and matches the one of its farthest region. (right) Gaussian map of the Euclidean farthest Voronoi diagram of a different set of segments in \mathbb{R}^3	10
2.5	The order-2 Voronoi diagram (in red) of three segments s_1, s_2, s_3 in the plane.	11
2.6	Point-hyperplane duality applied to segments: (left) Segments in primal space; and (right) their corresponding wedges in dual space	15
2.7	A cell complex in which none of the cells is unbounded in a specific direction.	16
2.8	An order-2 Voronoi diagram $VD_2(\{s_1, s_2, \dots, s_5\})$ (left) and its Gaussian map (right).	17
2.9	The farthest regions contain rays (left) and no farthest region can split the $d-1$ skeleton of the farthest Voronoi diagram into 2 parts (right).	19
2.10	A supporting hyperplane P (in dashed black) of sites H (in red) in direction v	20
2.11	A bounded region in the order-4 Voronoi diagram (in red) of segments (in black).	22
2.12	Construction of the path $\hat{\xi}$	22

2.13	A tunnel in the order-4 Voronoi diagram (in red) of segments (in black).	23
2.14	An instance of 5 segments (left), which has one region $\text{reg}(\{s_1, s_2, s_3\})$, shown in blue, on the Gaussian map of the order-3 Voronoi diagram (right) with high complexity.	24
2.15	(a) The angular distance $\angle(v, \ell)$ between line ℓ and direction v . (b) $\text{GM}(\text{FVD})$ of four lines in \mathbb{R}^3 . The farthest regions of the lines are colored in different colors. Vertices of anomaly are shown with squared boxes; proper vertices with disks.	26
2.16	(a) Lines L and their (b) transformed segments $\tau(L)$ have identical (c) Gaussian maps $\text{GM}(\text{VD}_2(L)) = \text{GM}(\text{VD}_2(\tau(L)))$	28
2.17	Construction of the path $(\hat{\xi}_1, \hat{\xi}, \hat{\xi}_2)$ for lines as sites.	29
2.18	Example for $d = 2$ and $n = 3$: Three 0-dimensional unit spheres (blue, green, red) split the unit circle into 6 arcs.	31
2.19	Illustration of Lemma 2.5.4. (a) The sites E (1 vertical red line and 3 line segments) in 3-space; (b) $\text{GM}(\text{FVD}(E))$ in \mathbb{R}^3 ; (c) The sites E projected to the x - y -plane Q (d) $\text{GM}(\text{FVD}(\text{Proj}_Q(E)))$ in \mathbb{R}^2 , which corresponds to the equator of $\text{GM}(\text{FVD}(E))$	35
3.1	The bisector (red) of two line segments (black) in the plane. The red points indicate the ends of the bisector's pieces, i.e. rays, parabolic arcs, and line segments. The minimum distance of the points on the bisector to one of the sites is realized at the green square.	44
3.2	The trisector of three lines (red, green, yellow) consists of up to four branches. The colors along the trisector encode the distance to the lines. In this example, the distance function along one branch admits a local maximum, as the color coding changes from orange to yellow and back to orange.	45
3.3	Different events during the collapse algorithm for constructing the farthest Voronoi of line segments in \mathbb{R}^2	48
3.4	There cannot be a vertex v in the farthest Voronoi diagram of lines segments with two incident edges with increasing distance towards v	50

4.1	Two approximations (a) and (b) of a Euclidean farthest-site Voronoi diagram (c). The sites are three overlapping triangles. Their boundaries are visualized in individual colors, and their farthest regions are painted accordingly. The distance polygons used—a square in (a) and a regular 8-gon in (b)—are shown in the bottom-left corner.	55
4.2	Polyhedral distance induced by \mathcal{P} : $d_{\mathcal{P}}(x, q) = 2$ and $d_{\mathcal{P}}(x, q') = 0.7$	57
4.3	Illustrations of the proofs of Lemma 4.3.2 (left) and Lemma 4.3.3 (right).	61
4.4	(a) The $(d-1)$ -skeleton can be disconnected for non-disjoint sites. (b) A weighted farthest Voronoi diagram of three sites: The blue quadrangle has an additive weight of -1 , and the red pentagon has a multiplicative weight of $\frac{1}{2}$	63
5.1	The Fermat point of the 28 EU-capitals (pre-Brexit), highlighted with (x), along with three 28-ellipses of different radii. (a) The foci (capitals) are unweighted. (b) Each focus has the weight of the country's population. The source of the map is https://www.consilium.europa.eu	66
5.2	The resulting box subdivision of Fig. 5.1(a) for (a) the n -ellipses and (b) the Fermat point.	68
5.3	(a) Example that a <i>good</i> approximation of the Fermat point in sense (B) does not imply a <i>good</i> approximation in sense (A). (b) Analogous example for sense (C).	71
5.4	Different steps during the the execution of Algorithm 3. The dark red boxes cannot contain the Fermat point, whereas the light green boxes may contain it.	74
5.5	An example with 500 foci, showing that Weiszfeld's scheme does not solve the ε -approximation problem. The scheme stopped when $\ p_{i-1} - p_i\ \leq 1/10$, after 207 steps (blue points). The distance $\ x^* - p_{207}\ $ can be arbitrarily big ($\ x^* - p_{207}\ > 15$ in this case).	79
5.6	The case analysis of Algorithm 5. (a) $\square N(B) \subseteq B$, (b) $\square N(\frac{B}{10}) \cap \frac{B}{10} = \emptyset$, and (c) $\square N(\frac{B}{10}) \cap \frac{B}{10} \neq \emptyset$	80
5.7	(a) A 3-ellipse passing through two foci. Components of gray boxes (temporarily) surround the foci. (b) If a gray component satisfies (B1) - (B3) the two ingoing edges are connected with an edge (shown dashed).	82

5.8	Two different 3-elliptic contour plots with 10 contour lines, having the same set of foci. (a) Using radii of <i>equidistant points</i> . (b) Using <i>equidistant radii</i>	85
5.9	A box subdivision for $n = 200$ foci: (a) UNIF-1, (b) UNIF-2 and (c) UNIF-2 after PCA.	86
5.10	A comparison of Algorithm 3 (● SUB), Algorithm 4 with the Krawczyk Newton operator (◇ Krawczyk), and Algorithm 4 with the Nickel and Moore Newton operator (■ Nickel & Moore). (a),(b) Time as a function of n , with $\varepsilon = 10^{-4}$. (c),(d) Time as a function of ε with $n = 100$. (a),(c) UNIF-1 datasets. (b),(d) UNIF-2 datasets.	87
5.11	Foci sets of TSPLib used in our experiments. burma14: 14 cities in Burma, bayg29: 29 cities in Bavaria, berlin52: 52 locations in Berlin, bier127: 127 beer gardens in the Augsburg area (Germany), tsp225: writing of TSP with 225 points, linhp318: 318 cities, ali535: 535 airports around the globe, nrw1379: Nordrhein-Westfalen (Germany), fnl4461: the five Federal States of Germany (ex-GDR territory), usa13509: cities in the continental US with at least 500 population	88
5.12	An overall comparison of Algorithm 3 (● SUB), Algorithm 4 with the PCA (◇ ESUB), and Algorithm 5 (■ CW). (a),(b) Time as a function of n , with $\varepsilon = 10^{-4}$. (c),(d) Time as a function of ε with $n = 100$. (a),(c) UNIF-1 datasets. (b),(d) UNIF-2 datasets.	89
5.13	(a) 100 points in convex position (b) points of a regular 100-gon (c) 100 points split among 10 clusters	90
5.14	(a) A comparison of TSP data sets (filled shapes) with UNIF-1 (empty shapes, dashed curve) for both Algorithm 4 (● ESUB) and Algorithm 5 (■ CW). Fermat point with time as a function of n . (b),(c) n -ellipse on UNIF-2 with time as a function of (b) n and (c) the length of the n -ellipse. Two ε approximations with $\varepsilon = 0.1$ (●) and $\varepsilon = 0.01$ (■) have been computed.	91
6.1	Red and blue points sets of cardinality 4 with high discrepancy. The indicated rectangle contains only red points, even though the expected difference between red and blue points in such a rectangle would 0.	97
6.2	The discrepancy of the red rectangle is roughly $ 8 \times 0.75^2 - 4 = 0.5$.	98
6.3	Illustration of f_j for $m = 3$ and $j = 1$	100
6.4	A Staircase	105

-
- 6.5 Staircase approximation for $m = 7$. The curves C_i are drawn in black. The brightness of the green color encodes the value of the discrepancy $D_{R,B}(x, y)$ at each point (x, y) . The discrepancy values range between -1 and 1 as shown on the right hand side. The staircases can be seen at the discontinuity of the discrepancy function. At each staircase the discrepancy function changes its value by 1. 106
- 6.6 An example of a CDR in one quadrant. 110
- 6.7 (left) A drawing of a CDR in $\mathcal{G}_N^+ \subset \mathbb{Z}^3$ for $N = 4$. Notice that the CDR is a tree whose leaves are at the plane $x + y + z = N$. (middle) A cross section on the xy -plane of the same CDR. Observe that vertices A and B do not extend within the xy -plane. Thus, the subspace is a weak CDR (rather than a proper CDR). (right) A map of the weak CDR into a two-colored point set. Regions with many blue points and few red correspond to portions of the CDR with high error. 111

Chapter 1

Introduction

1.1 Computational geometry in high dimensions

Geometric problems in higher dimensions induce multiple difficulties in contrast to their planar counterparts. The problem settings are harder to visualize and therefore finding useful properties can be challenging. Moreover, the algebraic complexity of the underlying formulas typically increases. Compare for example the algebraic description of edges in a Voronoi diagram of lines in two and three dimensions. In the plane, the bisector of three lines is just another line. In contrast, in 3-space the trisector of three lines is the solution of a polynomial of already degree four. The dimension of the space of interest does not need to be high and explicit exact representations of the basic objects are not possible to derive. Many concepts of data structures or algorithmic techniques work well in smaller dimensional spaces, but become inefficient once the dimension increases. An example is the quadtree/octree data structure. Generalizations of it to higher dimensions immediately induce an exponential dependency on the dimension. Finally the combinatorial complexity of the solution of the geometric problem often increases with the dimension and an algorithm to solve it needs to take care of that.

In the beginning of the research area of computational geometry, there was a big focus on the computation of convex hulls of points. Early algorithms for this problem in the plane were “Jarvis march” Jarvis [1973], and “Graham scan” Graham [1972]. There is a close relation between convex hulls and Voronoi diagrams of points, if one can compute the convex hull of points in \mathbb{R}^{d+1} then one can also derive the Euclidean Voronoi diagram of points in \mathbb{R}^d , see Edelsbrunner et al. [1989]. This is one of many examples, why the basic convex hull algorithms are so important. It took 20 years until an optimal deterministic time algorithm

for finding the convex hull of points in arbitrary dimensions was found Chazelle [1993]. In this thesis we discuss 3 topics in higher dimensions, described in the next section.

1.2 Dissertation Overview

In Chapter 2, we study the behavior at infinity of the farthest and the higher-order Voronoi diagram of n line segments or lines in a d -dimensional Euclidean space. The unbounded parts of these diagrams can be encoded by a *Gaussian map* on the sphere of directions \mathbb{S}^{d-1} . We show that the combinatorial complexity of the Gaussian map for the order- k Voronoi diagram of n line segments and lines is $O(\min\{k, n-k\}n^{d-1})$, which is tight for $n-k = O(1)$. All the d -dimensional cells of the farthest Voronoi diagram are unbounded, its $(d-1)$ -skeleton is connected, and it does not have tunnels. A d -cell of the Voronoi diagram is called a tunnel if the set of its unbounded directions, represented as points on its Gaussian map, is not connected. In a three-dimensional space, the farthest Voronoi diagram of lines has exactly $n^2 - n$ three-dimensional cells, when $n \geq 2$. The Gaussian map of the farthest Voronoi diagram of line segments and lines can be constructed in $O(n^{d-1}\alpha(n))$ time, while if $d = 3$, the time drops to worst-case optimal $O(n^2)$.

In Chapter 3, we review an $O(n \log n)$ time algorithm in the plane and describe obstacles for generalizing it for three dimensional space. In particular, we show that the distance function along a trisector of lines may have a local maximum.

In Chapter 4, we give asymptotically tight upper and lower worst-case bounds on the combinatorial size of the farthest Voronoi diagram for convex polyhedral distance functions with convex polyhedral sites in general dimensions, and propose an optimal time construction algorithm.

In Chapter 5, we present algorithms to compute an ε -approximation of the Fermat point \mathbf{x}^* . Our algorithms are based on the *subdivision* paradigm, which we enhance with Newton methods, used for *certification*, in the sense of *interval methods*, and for speed-ups. Moreover, we consider the problem of constructing n -ellipses, which are the r -level sets $\varphi^{-1}(r)$. Using the subdivision paradigm, we design an ε -isotopic approximation algorithm to compute n -ellipses in \mathbb{R}^2 . We have implemented our algorithms and we provide an experimental analysis using different point configurations and heuristics for speed-ups. The obtained results suggest the practicality of our approaches especially in low dimensions and for small epsilon.

In Chapter 6, we discuss the question of how uniform can one distribute red and blue points in the unit square, such that the difference between the number

of points in a sample area is close to the expected difference. We introduce the *bichromatic discrepancy* for point sets of two different colors within the unit square. We derive a lower bound for this bichromatic discrepancy and show its implications for consistent digital rays, a concept of digitalizing segments such that they still exhibit most Euclidean axioms. We also describe point sets which induce a low bichromatic discrepancy.

1.3 List of publications

Chapter 2 is based on the following paper:

- Gill Barequet, Evanthia Papadopoulou, and Martin Suderland. [2019]
Unbounded regions of high-order Voronoi diagrams of lines and segments in higher dimensions.
30th International Symposium on Algorithms and Computation (ISAAC 2019), Vol. 149 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 62:1–62:15.

Chapter 4 is based on the following paper:

- Franz Aurenhammer, Evanthia Papadopoulou, and Martin Suderland. [2021]
Piecewise-linear farthest-site Voronoi diagrams.
32nd International Symposium on Algorithms and Computation (ISAAC 2021), Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Chapter 5 is based on the following paper:

- Kolja Junginger, Ioannis Mantas, Evanthia Papadopoulou, Martin Suderland, and Chee Yap. [2021]
Certified approximation algorithms for the Fermat point and n -ellipses.
29th Annual European Symposium on Algorithms, ESA 2021, Vol. 204 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 54:1–54:19.

Chapter 6 is based on parts of the following paper:

- Man-Kwun Chiu, Matias Korman, Martin Suderland, and Takeshi Tokuyama. [2022]
Distance bounds for high dimensional consistent digital rays and 2-d partially-consistent digital rays.
Discrete & Computational Geometry pp. 1–43.

Chapter 2

Voronoi diagrams of generalized sites

This chapter is based on the following paper:

- Gill Barequet, Evanthia Papadopoulou, and Martin Suderland. [2019] Unbounded regions of high-order Voronoi diagrams of lines and segments in higher dimensions. *30th International Symposium on Algorithms and Computation (ISAAC 2019)*, Vol. 149 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 62:1–62:15.

The nearest Voronoi diagram of a set of geometric objects, called sites, is a well-known geometric partitioning structure. It partitions the relevant space into regions such that all points within a region have the same nearest site. Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of sites in d -dimensional space. We will only consider points, lines, line segments and convex polyhedra as sites. We denote with $d(x, s)$ the distance of a point $x \in \mathbb{R}^d$ to the site s . The distance can be the Euclidean distance or a distance induced by any convex polyhedron P which contains the origin. Note that the L_∞ distance can also be represented as a polyhedral distance.

The Voronoi region of site s_i of the nearest Voronoi diagram is defined as

$$\text{reg}(s_i) = \{ p \in \mathbb{R}^d \mid \forall j \neq i : d(p, s_i) \leq d(p, s_j) \}. \quad (2.1)$$

A maximally connected i -dimensional set of points, which is on the boundary of the same set of regions, is called an i -dimensional cell of the diagram, in short i -cell. In the special cases $i = 0, 1, 2, d$ we call an i -cell a vertex, edge, face or cell. The complexity of a Voronoi diagram is the total number of its i -cells for all $0 \leq i \leq d$ respectively. Given that each site is a connected object, the regions of the nearest diagram are connected and non-empty. A Euclidean nearest Voronoi diagram of points and segments is shown in Figure 2.1.

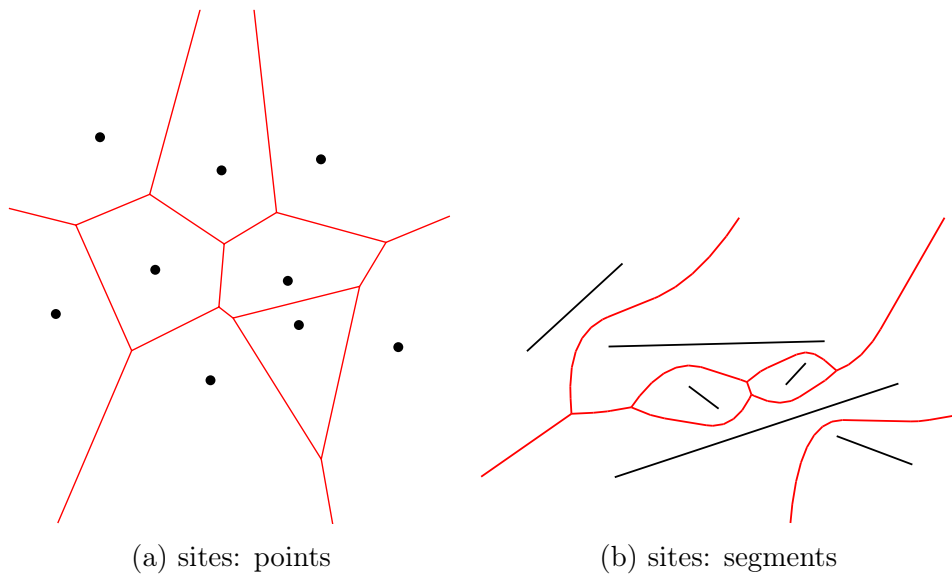


Figure 2.1. Euclidean nearest Voronoi diagram

The Voronoi diagram finds applications in diverse areas, when proximity information is important. Given n points in the plane, the Voronoi diagram can be used to find the Euclidean minimum spanning tree, the smallest circle enclosing the points, k nearest and farthest neighbours, the two closest points, the well-known Delaunay triangulation Shamos and Hoey [1975], clustering point sites and motion planning Aurenhammer [1991]. It is applied in economics, metallurgy, archaeology, ecology, geography, geology, astronomy and many more areas, see the book of Okabe et al. [1992].

The most extensively studied type of Voronoi diagram is the Euclidean Voronoi diagram of points. Edelsbrunner and Seidel [1986] noticed that in order to compute the Euclidean Voronoi diagram of points in \mathbb{R}^d it suffices to construct the convex hull of a set of points in \mathbb{R}^{d+1} . Chazelle [1991] finalized this algorithm by providing an optimal convex hull algorithm, which runs in $O(n \log n + n^{\lfloor \frac{d}{2} \rfloor})$ time given n points in \mathbb{R}^d . Combining these two results gives an $O(n \log n + n^{\lfloor \frac{d}{2} \rfloor})$ time algorithm to construct the Euclidean Voronoi diagram of points in \mathbb{R}^d . A matching worst-case lower bound on the complexity of the Euclidean Voronoi diagram of n points in \mathbb{R}^d of $\Omega(n^{\lfloor \frac{d}{2} \rfloor})$ was given in Klee [1980]. The necessity of the term $O(n \log n)$ follows from a simple reduction from sorting. Concerning the Voronoi diagram of points under certain polyhedral norms, the complexity is $\Theta(n^{\lfloor \frac{d}{2} \rfloor})$ for the L_∞ and L_1 distance and also for polyhedral distances induced by a d -simplex Boissonnat et al. [1998].

If we consider sites that are not points, things become more complicated.

Analysing lines and line segments seem to be a natural generalization of point sites. Moreover the Voronoi diagram of polygons can be constructed once we can deal with the Voronoi diagram of line segments. Many algorithmic paradigms, such as plane sweep, incremental construction or divide and conquer have been applied to construct the Voronoi diagram of line segments in the plane in optimal $O(n \log n)$ time Aurenhammer et al. [2013].

Already in three dimensional space the algebraic description of the features of the Voronoi diagram, especially the 1-cells, become quite complicated. The 37 pages paper Everett et al. [2009] describes only the Euclidean Voronoi diagram of just 3 lines. In special cases these diagrams have a simpler structure. For instance, the Voronoi diagram of segments with at most c distinct orientations has $O(c^4 n^{2+\epsilon})$ complexity where ϵ is any positive number, see Koltun and Sharir [2002]. Another special case is a set of parallel halflines as sites which was studied very recently by Aurenhammer et al. Aurenhammer et al. [2017]. Then the Voronoi diagram can be computed in $O(k \log(n))$ time where the number of faces k is bounded by $O(n^{2+\epsilon})$.

The complexity of nearest Voronoi diagrams of n lines in \mathbb{R}^3 under a polyhedral distance functions, induced by a convex polyhedra of constant complexity, can be bounded by $O(n^2 \alpha(n) \log(n))$ and $\Omega(n^2 \alpha(n))$, see Chew et al. [1998]. Here $\alpha(n)$ is the extremely slowly growing inverse Ackermann function. If we consider disjoint polyhedra as sites with n vertices in total, then the complexity is $O(n^{2+\epsilon})$ shown in Koltun and Sharir [2004], for any $\epsilon > 0$. If the sites are line segments then they sharpen the bound to $O(n^2 \alpha(n) \log(n))$.

Farthest-site Voronoi diagrams are useful for performing farthest neighbor queries among the sites, for computing the smallest ball that contacts all sites, and for finding the largest gap to be bridged between any two sites—to name a few or their applications. In contrast to the nearest diagram, not much research has been conducted on the farthest Voronoi diagram. It is a partition of the space into regions such that the points of one region have the same farthest site. We only have to slightly change Equation (2.1) in order to define the farthest regions of the diagram.

$$\text{freg}(s_i) = \{ p \in \mathbb{R}^d \mid \forall j \neq i : d(p, s_i) \geq d(p, s_j) \}. \quad (2.2)$$

A Euclidean farthest Voronoi diagram of points and segments is shown in Figure 2.2. The farthest Voronoi diagram has different structural properties than the nearest one. Seidel Seidel [1987] derived exact bounds on the maximal complexity of the Euclidean farthest Voronoi diagram of points in \mathbb{R}^d . Asymptotically the worst-case complexity grows with $\Theta(n^{\lfloor \frac{d}{2} \rfloor})$.

We are mostly interested in the farthest Voronoi diagram of lines, line segments

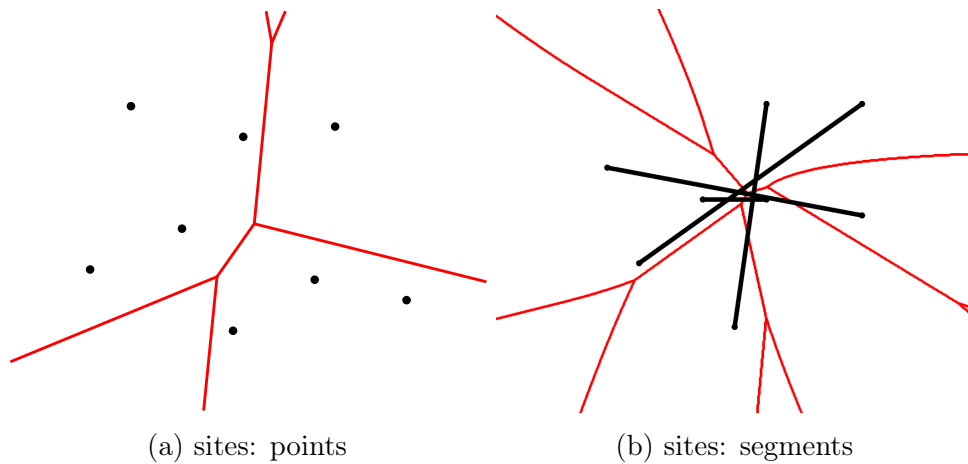


Figure 2.2. Euclidean farthest Voronoi diagram

or convex polyhedra under different distance measures in \mathbb{R}^d . We want to analyse its complexity and also look for efficient algorithms to construct the exact diagram or an approximation.

In two dimensions the Euclidean farthest Voronoi diagram of n segments has already been studied in Aurenhammer et al. [2006]. Aurenhammer et al. have shown that one farthest region can have up to $\Theta(n)$ many faces but the total complexity of the whole diagram still remains $O(n)$. They also gave an algorithm to compute the diagram in optimal $O(n \log n)$ time. Papadopoulou and Dey [2013] gave an output-sensitive $O(n \log h)$ time algorithm to construct the farthest line segment Voronoi diagram in the plane under any L_p metric with $1 \leq p \leq \infty$, where h is the number of unbounded faces. If segments intersect, then the number of intersections affects the complexity only if $k < \frac{n}{2}$ Papadopoulou and Zavershynskiy [2016]. These diagrams illustrate fundamental structural differences from their counterparts of points, such as disconnected Voronoi regions and no relation to convex hulls. Naturally, these differences carry over to higher dimensions.

Cheong et al. [2011] analysed the farthest Euclidean polygonal Voronoi diagram in \mathbb{R}^2 and proved that the complexity is $O(n)$ and that it can be computed in $O(n \log^3 n)$ time, assuming that the polygons are disjoint and have a complexity of n in total.

Already in three dimensional space with the Euclidean metric, there are no tight asymptomatic bounds on the complexity of the farthest Voronoi diagram known, similarly to the nearest. The only known bounds are the $\Omega(n^2)$ lower bound and a $O(n^{3+\epsilon})$ upper bound where ϵ can be any positive number Barequet

and Papadopoulou [2013]. The upper bound holds for a very general class of Voronoi diagrams in three dimensional space. It follows from the fact that the upper envelope¹ of n simple surfaces in \mathbb{R}^d has $O(n^{d-1+\epsilon})$ complexity Sharir [1994]. Computing a Voronoi diagram of n sites in d -dimensional space by computing an envelope in $d + 1$ dimensions of n so called distance functions is a common concept. The *distance function* of a site s is defined for each $x \in \mathbb{R}^d$ and maps x to its distance $d(x, s)$, see Fig. 2.3. The nearest neighbour Voronoi diagram corresponds to the lower envelope of the distance functions while the farthest Voronoi diagram corresponds to the upper envelope. Computing an envelope of non-linear surfaces is a challenging task which explains the difficulty in computing Voronoi diagrams of sites with non-linear distance functions.

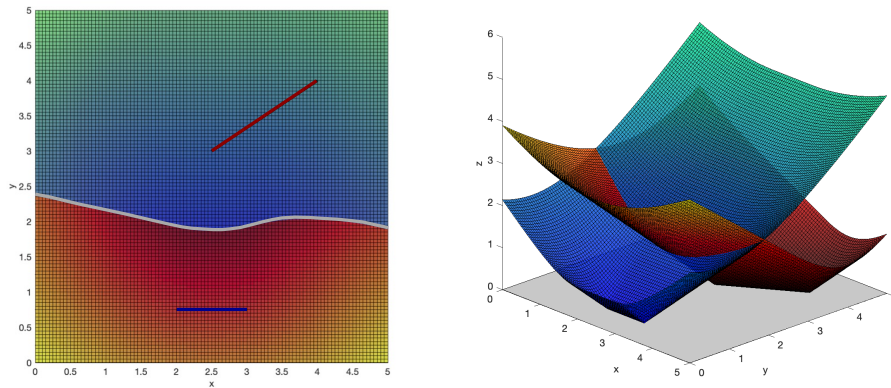


Figure 2.3. (left) The Euclidean farthest Voronoi diagram (in grey) of 2 line segments (red and blue). The intensity of the colors encode the proximity of a point to its closest site. Darker colors correspond to smaller distances. (right) The distance functions of both segments in \mathbb{R}^3 . The upper envelope of these distance functions correspond to the diagram on the left.

The Euclidean farthest Voronoi diagrams of lines and line segments has the property that all cells are unbounded see Barequet and Papadopoulou [2013]. This property motivates to first study the unbounded parts of the farthest Voronoi diagram. Barequet and Papadopoulou [2013] introduce a structure on a sphere of directions, called the Gaussian Map (in short GMap), describing those unbounded parts. Each direction on the sphere is associated with the site whose region is unbounded in that direction. This results in a subdivision of the sphere of directions

¹The upper (resp. lower) envelope of a set of surfaces is the pointwise maximum (resp. minimum) of all surfaces.

into regions as well, see Figures 2.4 and 2.4. They showed that the complexity of the GMap in \mathbb{R}^3 can be as high as $\Theta(n^2)$, see Barequet and Papadopoulou [2014].

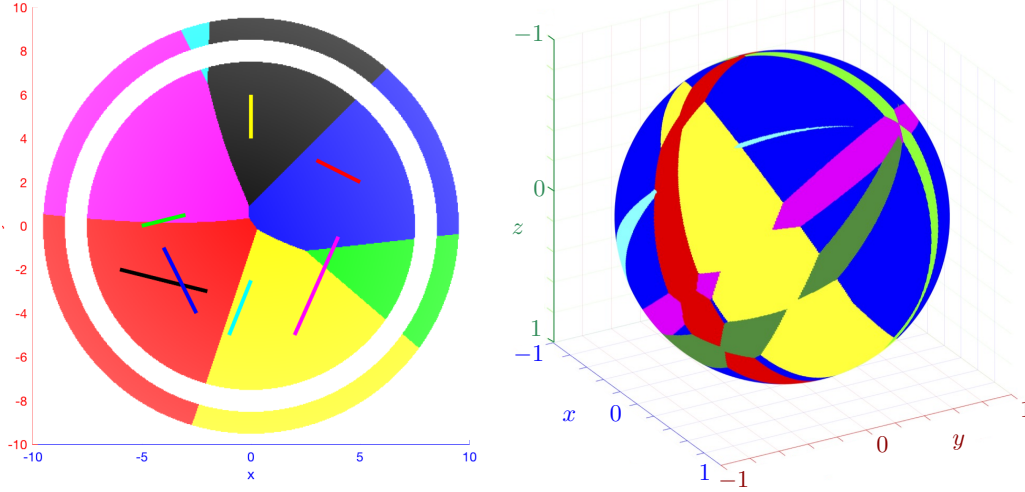


Figure 2.4. (left) The Euclidean farthest Voronoi diagram of segments in \mathbb{R}^2 and the intersection with a large circle, which approximates its Gaussian map. Each segment has a different color and matches the one of its farthest region. (right) Gaussian map of the Euclidean farthest Voronoi diagram of a different set of segments in \mathbb{R}^3 .

A generalization of the nearest and farthest Voronoi diagram are the higher-order Voronoi diagrams. For a subset of sites $H \subset S$ of cardinality k , the *order- k region* of H is the set of points in \mathbb{R}^d whose distance to any site in H is smaller than to any site not in H .

$$\text{reg}(H) = \{ p \in \mathbb{R}^d \mid \forall h \in H \forall s \in S \setminus H : d(p, h) \leq d(p, s) \}$$

When $k = 1$ (resp. $k = n - 1$), this diagram is the nearest-neighbor Voronoi diagram. For $k = n - 1$, it is the *farthest site Voronoi diagram*, denoted by $\text{FVD}(S)$.

The order- k Voronoi diagram corresponds to the k -th level of the arrangement of distance functions of the sites Edelsbrunner and Seidel [1986]. For points in the plane as sites with the Euclidean distance measure, the complexity of the diagram is bounded by $O(k(n - k))$ Lee [1982] and there are various algorithms with running times depending differently on n and k such as $O(k^2 n \log n)$ or $O(k(n - k) \log n + n \log^3 n)$ Lee [1982]; Chazelle and Edelsbrunner [1985]; Aurenhammer [1990]; Clarkson [1987]; Agarwal et al. [1998].

The diagram has also been studied for non-point sites. Papadopoulou and Zavershynskiy [2016] proved a similar complexity result for line segments. If

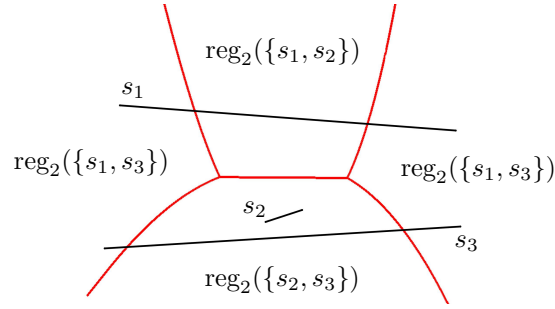


Figure 2.5. The order-2 Voronoi diagram (in red) of three segments s_1, s_2, s_3 in the plane.

$k < n/2$, then the complexity is bounded by $O(k(n-k) + I)$, where I is the number of line segment intersections. Surprisingly the number of intersections does not matter if $k \geq n/2$ and the complexity is simply bounded by $O(k(n-k))$, which is the same bound for points as sites. The diagram can be constructed by an iterative approach in $O(k^2 n \log n)$ time by first computing the nearest Voronoi diagram and then using the an order- i diagram to compute an order- $(i+1)$ diagram. In the context of abstract Voronoi diagrams the number of faces F_k of the order- k version was bounded by $n-k+1 \leq F_k \leq 2k(n-k) + k + 1 - n$ Bohler et al. [2015]. Abstract Voronoi diagrams are a unifying concept that include many concrete Voronoi diagrams Klein [2016]. Their definition relies on sets of bisecting curves instead of actual sites. These bisectors need to satisfy some conditions to fall under the umbrella of abstract Voronoi diagrams, such as that the curves are Jordan curves, each resulting Voronoi region is connected and the set of all regions covers the plane.

The closed order- k regions of S form a subdivision of \mathbb{R}^d . The induced cell complex is called the *order- k Voronoi diagram* of S , denoted by $\text{VD}_k(S)$. A maximally connected i -dimensional set of points, which is on the boundary of the same set of order- k regions, is called an i -dimensional cell of the cell complex. We call the i -dimensional cells of the order- k Voronoi diagram “ i -cells.”

When $k = 1$, this diagram is the well-known nearest-neighbor Voronoi diagram, denoted by $\text{VD}(S)$. For $k = n - 1$, it is the *farthest site Voronoi diagram*, denoted by $\text{FVD}(S)$. Its *farthest regions* can also be defined directly as $\text{freg}(h) = \{p \in \mathbb{R}^d \mid s \in S \setminus \{h\} : d(p, h) > d(p, s)\}$.

Another generalization is the addition of weights to the sites. The weights describe the influence of a site to its nearby points. For example, the regions of

the weighted nearest Voronoi diagram are defined as

$$\text{reg}(s_i) = \left\{ p \in \mathbb{R}^d \mid \forall j \neq i : \frac{d(p, s_i)}{w_{s_i}} - W_{s_i} \leq \frac{d(p, s_j)}{w_{s_j}} - W_{s_j} \right\} \quad (2.3)$$

where we assign to each site s a multiplicative weight w_s and an additive weight W_s . Weights can also be added to all other mentioned versions Voronoi diagrams in a similar fashion. In the nearest (resp. farthest) Voronoi diagram a site's region is expanding (resp. shrinking) with increasing weight. Euclidean (additively and multiplicatively) weighted Voronoi diagrams in \mathbb{R}^d can be modeled as a so called "power diagrams" in \mathbb{R}^{d+1} Aurenhammer [1987]. Power diagrams in \mathbb{R}^{d+1} can be constructed through a convex hull algorithm in \mathbb{R}^{d+2} in $O(n \log n + n^{\lfloor \frac{d}{2} \rfloor + 1})$ time.

Summarizing there are several variations of the standard Euclidean Voronoi diagram of points, i.e. using different sites, weights, distances or working with higher-order diagrams in higher dimensions.

In the plane, many algorithmic paradigms, such as plane sweep, incremental construction, and divide-and-conquer have been applied to construct the Voronoi diagram of line segments Aurenhammer et al. [2013]. In higher-dimensional spaces, however, results are quite sparse. As a result, the combinatorial complexity of this diagram has been a major open problem in computational geometry Mitchell and O'Rourke [2001]. There is a gap of an order of magnitude between the $\Omega(n^2)$ lower bound Aronov [2002] and the only known upper bound of $O(n^{3+\epsilon})$ Sharir [1994], where n is the number of sites. The gap carries over (and expands) to the Voronoi diagram of lines in d -space, $d \geq 3$, where the known bounds are $\Omega(n^{\lfloor \frac{d}{2} \rfloor})$ Aronov [2019] and $O(n^{d+\epsilon})$ Sharir [1994]. The lower bound is derived from n parallel lines whose Voronoi diagram has the same complexity as the Voronoi diagram of n points in $d-1$ dimensional space. For points in \mathbb{R}^d , the bound is $\Theta(n^{\lfloor \frac{d}{2} \rfloor})$ Aurenhammer et al. [2013], and for $(d-2)$ -dimensional hyperplanes, the lower bound is $\Omega(n^{d-1})$ Aronov [2002]. To the best of our knowledge, no other lower bound, other than $\Omega(n^{\lfloor \frac{d}{2} \rfloor})$, is available for line segments in \mathbb{R}^d , $d > 3$. Better combinatorial bounds are known only for some restricted cases Aurenhammer et al. [2017]; Chew et al. [1998]; Koltun and Sharir [2002, 2004]. A numerically robust algorithm for computing the Voronoi diagram of lines in 3D has been given by Hemmer et al. [2010].

In three dimensions, the Euclidean farthest-site Voronoi diagram of lines or line segments has the property that all its three-dimensional cells are unbounded Barequet and Papadopoulou [2013]. Barequet and Papadopoulou [2013] used a structure on the sphere of directions, called the Gaussian map, which reflects the directions under which the cells of this diagram are unbounded. The Gaussian

map essentially replaces the role of the convex hull in characterizing the unbounded regions in the higher-order Voronoi diagram of $\text{VD}_k(E)$, for $k > 1$. In this chapter, we study the Gaussian map of order- k and farthest Voronoi diagrams of n line segments and lines as sites in \mathbb{R}^d , and characterize the unbounded directions of the cells in these diagrams. The dimension d is assumed to be constant. We derive the bound $O(\min\{k, n-k\}n^{d-1})$ on the complexity of the Gaussian map of order- k Voronoi diagrams for these sites. This implies the same upper bound on the complexity of the unbounded features of the corresponding order- k Voronoi diagrams. For the farthest-site diagram ($k = n - 1$), this is $O(n^{d-1})$. For segments as sites, we prove that the complexity of the Gaussian map is $\Omega(k^{d-1})$, which is tight when $n - k = O(1)$. In fact, the complexity bound is derived from the number of vertices on the Gaussian map. This leads to a lower bound of $\Omega(k^{d-1})$ on the complexity of the entire order- k Voronoi diagram for line segments. For the farthest-site Voronoi diagram, this bound becomes $\Omega(n^{d-1})$, which also holds for lines as sites. As a byproduct, we derive a bound on the complexity of the arrangement of n great hyperspheres on \mathbb{S}^{d-1} .

Further, we describe a transformation that maps a set of lines to a set of segments, such that the two respective Gaussian maps of order- k Voronoi diagrams are identical. This transformation can be used to carry lower bounds from lines to segments and upper bounds from segments to lines. Table 2.1 summarizes most of the complexity results derived here. All the d -dimensional cells of the farthest Voronoi diagram of both lines and segments are unbounded, its $(d-1)$ -skeleton is connected, and it does not have tunnels. In three dimensions, the farthest Voronoi diagram of lines in general position has exactly $n(n-1)$ many 3-dimensional cells, when $n \geq 2$. We show that we can compute the Gaussian map of this diagram in $O(n^{d-1}\alpha(n))$ time by using the algorithm of Edelsbrunner et al. [1989], which extends to higher dimensions Agarwal and Sharir [2000]; Halperin and Sharir [2017], for computing the envelope of piecewise-linear functions in \mathbb{R}^d . In fact, we conjecture that this bound can be improved to $O(n^{d-1})$. In three dimensions, we can compute the Gaussian map of the farthest Voronoi diagram of lines or segments in $O(n^2)$ time, following Edelsbrunner et al. [1989], which is optimal in the worst-case.

The chapter is organized as follows. In Section 2.1 we give an introduction on the basic concepts used. In Section 2.2 we describe some basic properties of farthest and order- k Voronoi diagrams. Section 2.3 studies the Gaussian map of the order- k Voronoi diagram for a set of line segments as sites, including bounds on its complexity and also a worst-case optimal time algorithm. In Section 2.4 the close relation to lines as sites is discussed. Section 2.5 looks at combining segments and lines and finally Section 2.6 concludes with generalizations to

Structure	Lower bound	Upper bound
GM(VD _k (E))	$\Omega(k^{d-1})^*$	$O(\min\{k, n-k\}n^{d-1})$
GM(FVD(E))	$\Omega(n^{d-1})$	$O(n^{d-1})$
VD _k (E)	$\Omega(k^{d-1})^*$	$O(\min\{k, n-k\}n^{d+\varepsilon})$
FVD(E)	$\Omega(n^{d-1})$	$O(n^{d+\varepsilon})$

*Only for segments.

Table 2.1. Worst-case complexities of structures induced by a set E of n lines or segments in \mathbb{R}^d .

polyhedra or clusters of points as sites.

2.1 Preliminaries

2.1.1 Point-Hyperplane Duality

Under the well-known point-hyperplane duality T in \mathbb{R}^d , a point $p \in \mathbb{R}^d$ is transformed to a non-vertical hyperplane $T(p)$, and vice versa. The transformation maps a point with coordinates (p_1, p_2, \dots, p_d) to the hyperplane $T(p)$ satisfying the equation $x_d = -p_d + \sum_{i=1}^{d-1} p_i x_i$. The transformation is an involution, i.e., $T = T^{-1}$.

For a segment $s = uv$, the hyperplanes $T(u)$ and $T(v)$ partition the dual space into four *wedges*, among which the *lower wedge* (resp., the *upper wedge*) is the one that lies below (resp., above) both $T(u)$ and $T(v)$. The ridge of the wedge is the intersection of $T(u)$ and $T(v)$.

Let E be a set of n segments, which in dual space corresponds to an arrangement of lower wedges. Let L_k be the k th level of that arrangement. Let p be a point on L_k , which touches the dual wedge of segment $s = ab$, and let H be the set of segments whose wedges are below p , see Figure 2.6. Then, the point p corresponds to a hyperplane $T^{-1}(p)$ which touches the segment s . The closed halfspace above $T^{-1}(p)$ has a non-empty intersection with the segments in H . The open halfspace above $T^{-1}(p)$ does not intersect any segment in $E \setminus H$. We will use this property when we study the Gaussian map, which is defined in the next section.

2.1.2 Levels in an Arrangement of Hyperplanes

In this section, we review the definition of levels of an arrangement of surfaces, where those surfaces satisfy some *mildness* conditions (A1)-(A3) as given in

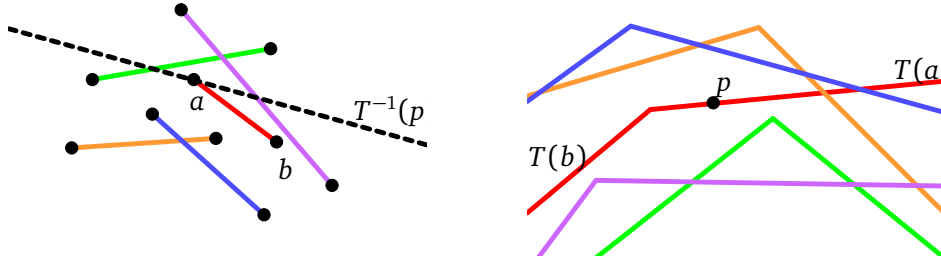


Figure 2.6. Point-hyperplane duality applied to segments: (left) Segments in primal space; and (right) their corresponding wedges in dual space

Agarwal and Sharir [2000]. We will use Theorem 2.1.1 by Clarkson and Shor several times. The level of a point $p \in \mathbb{R}^d$ in an arrangement $\mathcal{A}(\Gamma)$ of a set Γ of surface patches is the number of surfaces of Γ lying vertically below p . For $0 \leq k < n$, the k -level (resp., $\leq k$ -level), denoted by $\mathcal{A}_k(\Gamma)$ (resp., $\mathcal{A}_{\leq k}(\Gamma)$), is the closure of all points on the surface of Γ whose level is k (resp., at most k). A face of $\mathcal{A}_k(\Gamma)$ or $\mathcal{A}_{\leq k}(\Gamma)$ is a maximal connected portion of a face of $\mathcal{A}(\Gamma)$ consisting of points having a fixed subset of surfaces lying below them. Let $\psi_k(\Gamma)$ (resp., $\psi_{\leq k}(\Gamma)$) be the total number of faces in $\mathcal{A}_k(\Gamma)$ (resp., $\mathcal{A}_{\leq k}(\Gamma)$). Agarwal and Sharir [2000]

Theorem 2.1.1 (Clarkson and Shor [1989]). *Let \mathcal{G} be an infinite family of surfaces satisfying some mildness assumptions (A1)-(A3) described in Agarwal and Sharir [2000]. Then, for any $0 \leq k < n - d$,*

$$\psi_{\leq k}(n, d, \mathcal{G}) = O\left((k+1)^d \mathcal{C}\left(\frac{n}{k+1}, d, \mathcal{G}\right)\right),$$

where $\mathcal{C}(n, d, \mathcal{G})$ is the maximum complexity of the lower envelope of n surfaces in \mathcal{G} .

It obviously holds that $\psi_k(\Gamma) \leq \psi_{\leq k}(\Gamma)$.

2.1.3 The Gaussian Map

Let M be a cell complex in \mathbb{R}^d . The *complexity* of M is the total number of the cells of M of all dimensions. The *Gaussian map* of M encodes information about the unbounded cells of M . This structure is of particular interest when all d -cells of M are unbounded. For example, all the d -dimensional cells of the farthest Voronoi diagram of segments or lines are unbounded.

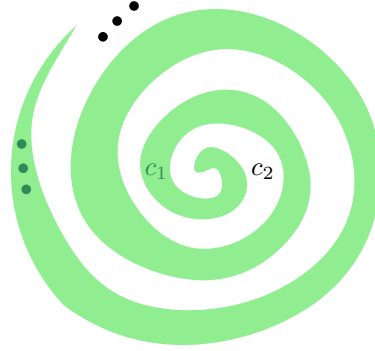


Figure 2.7. A cell complex in which none of the cells is unbounded in a specific direction.

Definition 2.1.2. A d -cell of M is unbounded in direction \vec{v} if it contains a ray with direction \vec{v} .

The idea of a cell containing a ray works well for defining a cell's unbounded directions if the cell is d -dimensional. However, it is not adequate for non-linear cells of dimension $< d$. For example, the trisector of three lines is in general a non-linear curve Everett et al. [2009], containing no ray. Imagine a point p moving along a branch of the trisector to infinity. The tangent of the trisector at p does actually converge and we want to call this direction an *unbounded* direction of the trisector. The next definition refines Definition 2.1.2 for cells of dimension $< d$.

Definition 2.1.3. A cell c of M is unbounded in direction \vec{v} if in the limit $\lambda \rightarrow 0$, the intersection of the scaled cell $\lambda \cdot c$ and the unit sphere \mathbb{S}^{d-1} is non-empty in direction \vec{v} .

The scaling of cell c can be done with an arbitrary center. Here, the limit $\lim_{\lambda \rightarrow 0} (\lambda c \cap \mathbb{S}^{d-1})$ should be understood with the concept of the Kuratowski convergence Kuratowski [1966], which we briefly review. For any point $x \in \mathbb{R}^d$ and subset $\mathcal{S} \subset \mathbb{R}^d$, let $d(x, \mathcal{S}) = \inf\{d(x, \sigma) \mid \sigma \in \mathcal{S}\}$ be the distance between x and \mathcal{S} . Let $\mathcal{S}_\lambda \subset \mathbb{R}^d$ be a compact set for any $\lambda > 0$. We say that \mathcal{S}_λ converges to \mathcal{S} for $\lambda \rightarrow 0$ if $\mathcal{S} = \{x \in \mathbb{R}^d \mid \limsup_{\lambda \rightarrow 0} d(x, \mathcal{S}_\lambda) = 0\} = \{x \in \mathbb{R}^d \mid \liminf_{\lambda \rightarrow 0} d(x, \mathcal{S}_\lambda) = 0\}$.

Note that the Kuratowski limit does not always exist Kuratowski [1966]. Consider a cell complex consisting of 2 cells circling around each other, see Fig. 2.7. The unbounded directions of the cells of this cell complex are not be defined in this

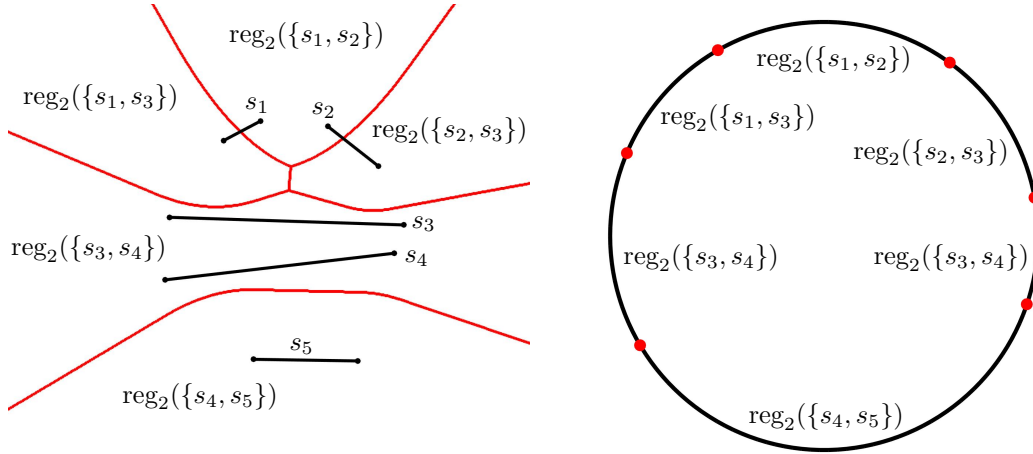


Figure 2.8. An order-2 Voronoi diagram $\text{VD}_2(\{s_1, s_2, \dots, s_5\})$ (left) and its Gaussian map (right).

case, because for any cell $c \in \{c_1, c_2\}$ the sets $\{x \in \mathbb{R}^d \mid \limsup_{\lambda \rightarrow 0} d(x, \lambda c \cap \mathbb{S}^1) = 0\} = \emptyset$ and $\{x \in \mathbb{R}^d \mid \liminf_{\lambda \rightarrow 0} d(x, \lambda c \cap \mathbb{S}^1) = 0\} = \mathbb{S}^1$ are not the same. We only consider cell complexes in which the unbounded directions of cells are well defined.

Definition 2.1.4. *The Gaussian map of M , denoted by $\text{GM}(M)$, maps each cell in M to its unbounded directions, which are encoded on the unit sphere \mathbb{S}^{d-1} , see Figure 2.8. Let c be a cell of M ; the set of directions, in which c is unbounded, is called the region of c on $\text{GM}(M)$. The part of $\text{GM}(M)$ where the d th coordinate is ≥ 0 (resp., ≤ 0) is called the upper (resp., lower) Gaussian map.*

The Kuratowski limit is a closed set, if it exists, and therefore, cells of the Gaussian map are closed.

We focus on cell complexes, such as the farthest Voronoi diagram and the order- k Voronoi diagram of lines and segments, where cells have unbounded directions and the Gaussian map is the respective partition of \mathbb{S}^{d-1} . This partition induces a cell complex on \mathbb{S}^{d-1} . The collection of cells on the Gaussian map of a Voronoi diagram $\text{VD}_k(E)$, which correspond to the same set of sites $H \subset E$, is called the *region of H* on $\text{GM}(\text{VD}_k(E))$.

A Gaussian map region of a set of sites may consist of several $(d-1)$ -cells for two reasons: A region of a set of sites of VD_k may split into many d -cells, which all have unbounded directions on the Gaussian map. Moreover, the Gaussian map region of just one d -cell of VD_k can consist of several $(d-1)$ -cells, e.g., $\text{reg}(\{s_3, s_4\})$ in Fig. 2.8.

Definition 2.1.5. A d -cell of the order- k Voronoi diagram is called a *tunnel* if its set of unbounded directions, represented as points on its Gaussian map, is not connected.

In Figure 2.8, the cell $\text{reg}(\{s_3, s_4\})$ forms a *tunnel* in $\text{VD}_2(E)$.

The Gaussian map essentially replaces the role of the convex hull in characterizing the unbounded regions in the higher-order Voronoi diagram of $\text{VD}_k(E)$, for $k > 1$.

2.2 Properties of the Farthest and Order- k Voronoi Diagram

2.2.1 Combinatorial Properties

It has already been stated Barequet and Papadopoulou [2013] that the complexity of the farthest Voronoi diagram is $O(n^{3+\varepsilon})$ by following the general bound of Sharir Sharir [1994]. This bound generalizes for the order- k Voronoi diagram in \mathbb{R}^d .

Theorem 2.2.1. *The order- k Voronoi diagram of segments and lines in \mathbb{R}^d has complexity $O(\min\{k, n-k\}n^{d+\varepsilon})$.*

Proof. Each site induces a distance function, which maps every point in \mathbb{R}^d to its distance to that site. The general framework of Sharir [1994] shows that the complexity of the 0-level (resp., $(n-1)$ -level) of those distance functions is $O(n^{d+\varepsilon})$. Applying Theorem 2.1.1 by Clarkson and Shor [1989], the complexity of the $\leq k$ -level is $O(kn^{d+\varepsilon})$ and $O((n-k)n^{d+\varepsilon})$. \square

In Section 2.3 we will prove the following lower bounds. These bounds are meaningful when k is comparable to n .

Theorem 2.2.2. *The complexity of the order- k Voronoi diagram of segments in \mathbb{R}^d is $\Omega(k^{d-1})$ in the worst case. For the farthest Voronoi diagram ($k = n-1$), this lower bound is $\Omega(n^{d-1})$.*

2.2.2 Structural Properties

Lemma 2.2.3. *Let E be a set of lines and segments, and let $p \in \text{freg}(e)$ be a point in the farthest region of site e . Let t be the point on e , which realizes the distance between e and p . Then, the entire ray \vec{r} , that emanates from p with direction \vec{tp} , is contained in $\text{freg}(e)$.*

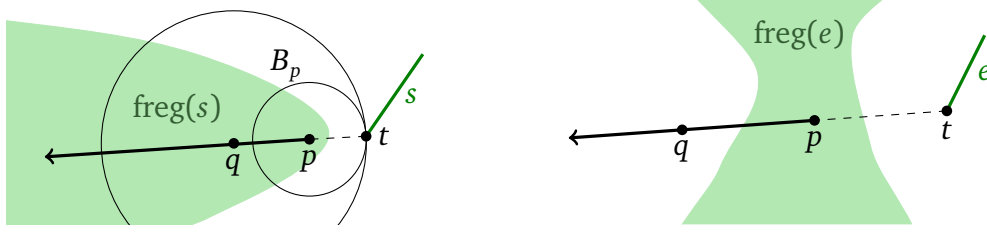


Figure 2.9. The farthest regions contain rays (left) and no farthest region can split the $d-1$ skeleton of the farthest Voronoi diagram into 2 parts (right).

Proof. The ball B_p , centered at p and of radius $|pt|$, touches e . Its interior intersects all other sites in E . In addition, any hyperball centered at any point $q \neq p$ along \overrightarrow{r} and of radius $|qt|$ must be properly enclosing B_p while touching e at t , see Fig. 2.9. Thus, it must also intersect all sites in E except e . Therefore, $\text{freg}(e)$ must contain the entire ray \overrightarrow{r} . \square

Corollary 2.2.4. *Let E be a set of lines and segments. All d -cells of $\text{FVD}(E)$ are unbounded.*

The VD_k of segments can have bounded regions if $d \leq k \leq n-2$. We defer the proof of Section 2.2.2 to Section 2.3.

Definition 2.2.5. *The i -skeleton of a cell complex M is the union of all j -cells in M with dimension $j \leq i$.*

Theorem 2.2.6. *Let E be a set of lines and segments in \mathbb{R}^d . The $(d-1)$ -skeleton of $\text{FVD}(E)$ is connected.*

Proof. Assume, for the sake of contradiction, that the diagram is not connected. Then, there exists a d -cell c that splits the $(d-1)$ -skeleton into at least two parts. Let e be the farthest site corresponding to c . The site e does not touch $\text{freg}(e)$. Let q be a point, which is separated from e by c . Let t be a point on e , which realizes the distance between q and e . Let p be a point on the segment \overline{qt} in $\text{freg}(e)$, see Fig. 2.9. Then, by Lemma 2.2.3, the entire ray \overrightarrow{r} , emanating from p in direction \overrightarrow{pq} , is contained in $\text{freg}(e)$. In particular, $q \in \text{freg}(e)$, which is a contradiction. \square

The $(d-1)$ -skeleton of $\text{VD}_k(E)$ need not be connected for $k \leq n-2$ and E being a set of sites in the plane, see Fig. 2.8.

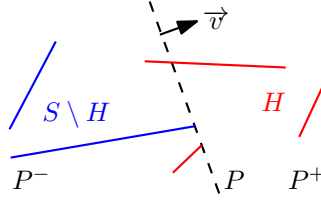


Figure 2.10. A supporting hyperplane P (in dashed black) of sites H (in red) in direction v .

2.3 Line Segments as Sites

Let E be a set of line segments in \mathbb{R}^d . We assume that the segments are in general position, *i.e.*, no $(d+1)$ segment endpoints lie on the same hyperplane. First, we characterize the segments that induce unbounded regions in the order- k Voronoi diagram in a given direction \vec{v} .

Definition 2.3.1. Let E be a set of sites, and let H be a subset of E . A hyperplane P is called a supporting hyperplane of H in direction \vec{v} if

1. P is orthogonal to \vec{v} ;
2. The closed halfspace P^+ , bounded by P and unbounded in direction \vec{v} , intersects each of the sites in H ; and
3. The sites in $E \setminus H$ do not intersect the interior of P^+ , and at least one site in $E \setminus H$ touches P .

Figure 2.10 illustrates a hyperplane supporting three segments.

The following theorem is a generalization of results for the plane Aurenhammer et al. [2006]; Papadopoulou and Zavershynskiy [2016].

Theorem 2.3.2. A set of segments H , with $|H| = k$, induces an unbounded region in direction \vec{v} in the order- k Voronoi diagram of segments S , if and only if there exists a supporting hyperplane of H in direction \vec{v} .

Proof. Let H be a set of k segments, which has an unbounded d -cell c in direction \vec{v} in the order- k Voronoi diagram of a set of segments S . Each point in the cell corresponds to the center of a closed ball which has non-empty intersection with the segments in H , and does not intersect any of the other segments. By definition, there exists a curve unbounded in direction \vec{v} , which is contained in c . Any point p on that curve is the center of a closed ball, which has a non-empty intersection with the segments in H and does not intersect any of the

other segments in its interior. When p moves along the curve to infinity, the ball around p becomes a halfspace which is orthogonal to \vec{v} . By moving the bounding hyperplane in direction $-\vec{v}$ until it hits a segment in $S \setminus H$, we can make it a supporting hyperplane.

Let P be a supporting hyperplane of segments H in direction \vec{v} . Let $H' \subseteq H$ be the subset of segments in H that touch P . Let x be a point on P , which is closer to all endpoints of segments in H' than those which belong to other segments. Consider the ray r which emanates from x and is unbounded in direction \vec{v} . On that ray, we find a point y , which is the center of a closed ball, which touches x and intersects only the segments in H . Every point z on r beyond the point y has the same properties because the ball keeps growing on the side P^+ and shrinks on the other side. This means that all those points on r beyond y belong to the order- k region of the set H . \square

Corollary 2.3.3. *A supporting hyperplane of H in direction \vec{v} , which touches i segments (at least one of which is in H), corresponds to a $(d-i+1)$ -cell in $VD_k(S)$, which is unbounded in direction \vec{v} , and to a $(d-i)$ -cell in $GM(VD_k(S))$.*

Proof. The hyperplane is supporting several sets of sites of cardinality k at the same time. The farthest regions of all those sets are unbounded in direction \vec{v} . Moreover, these regions are adjacent to the locus of points that are equidistant from those i sites touching the hyperplane. This locus of points is $(d-i+1)$ -dimensional and is unbounded in direction v , because all its adjacent regions are unbounded in v .

In order to work out the dimension of the Gaussian map cell in direction \vec{v} , we need to analyze the directions in which we can rotate the hyperplane P , such that it will still support the exact same set of sites. In particular, the sites which touch the hyperplane originally need to keep touching it during the rotation. If the hyperplane were touching only one endpoint of a segment, then the rotation would have $d-1$ degrees of freedom. With each additional endpoint, which touches the hyperplane, the number of degrees of freedom reduces by 1. \square

Note that Corollary 2.3.3 and its proof generalize to convex polyhedra and clusters of points as sites, see Section 2.6. We can now provide a proof deferred from Section 2.2.2.

Proof of Section 2.2.2. Indeed, we can provide a construction for points, which can easily be extended to segments if we replace the points by small-enough segments. Put k points close to each other, such that there exists a point m in the interior of their convex hull. Add two more far-apart points, such that m is their

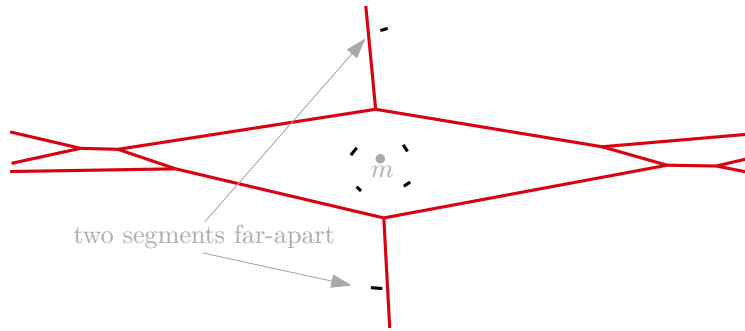


Figure 2.11. A bounded region in the order-4 Voronoi diagram (in red) of segments (in black).

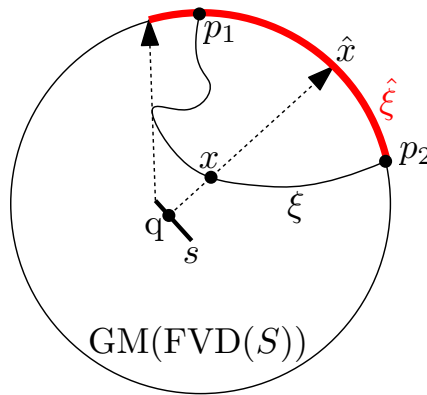


Figure 2.12. Construction of the path $\hat{\xi}$.

midpoint, see Fig. 2.11. Then, the k points have a d -cell in the order- k Voronoi diagram. On the other hand, those k points do not admit a supporting hyperplane and, therefore, cannot have an unbounded d -cell by Theorem 2.3.2. It is possible to add additional points to this configuration, as long as their distance to the first k points is big enough. \square

Theorem 2.3.4. *Let S be a set of segments. Then, $FVD(S)$ does not have tunnels.*

Proof. Let $p_1, p_2 \in GM(FVD(S))$ be two points representing unbounded directions of a farthest cell of segment s . These two points represent directions \vec{r}_1, \vec{r}_2 along which there exist points $x_1, x_2 \in \text{freg}(s)$, for which $\vec{r}_i = \overrightarrow{q_i x_i}$ with $(i = 1, 2)$, where q_i is the point on s realizing the distance between x_i and s . Since x_1 and x_2 are contained in the same cell of $\text{freg}(s)$, there exists a continuous path ξ connecting the points and being fully contained in $\text{freg}(s)$. We can map every point $x \in \xi$ to the direction $\vec{r} = \overrightarrow{q x}$, with q realizing the distance between x

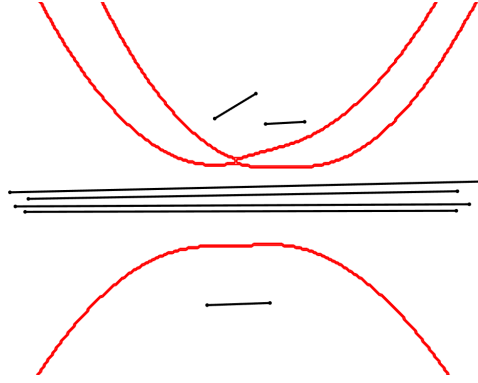


Figure 2.13. A tunnel in the order-4 Voronoi diagram (in red) of segments (in black).

and s . We represent direction \vec{r} as a point $p \in \text{GM}(\text{FVD}(S))$. Note that p is contained in a farthest cell of the Gaussian map corresponding to segment s . By continuity, mapping the whole path ξ to $\text{GM}(\text{FVD}(S))$ draws a continuous path $\hat{\xi}$ between p_1 and p_2 consisting solely of points that belong to s . Therefore, the points p_1 and p_2 belong to the same cell of the Gaussian map. \square

The order- k Voronoi diagram of segments S can have tunnels, for $k \leq n - d$.

Proof. Take k long segments very close to each other and almost parallel, and put at least d small segments close to and around their midpoints, see Fig. 2.13. Then, the order- k region of the k long segments creates a tunnel. \square

The next theorem provides a lower bound on the complexity of the Gaussian map of order- k Voronoi diagrams. This bound is meaningful if k is comparable to n .

Theorem 2.3.5. *Let S be a set of n line segments in \mathbb{R}^d . A single region of the Gaussian map of the order- k Voronoi diagram of S can have $\Omega(k^{d-1})$ many vertices.*

Proof. The bound is shown by a generalization of examples provided for \mathbb{R}^2 Aurenhammer et al. [2006]; Papadopoulou and Zavershynskiy [2016]. Place k long segments connecting almost antipodal points on a $(d-1)$ -dimensional hypersphere and $n-k$ additional short segments near the center of the hypersphere, see Figure 2.14. Any $(d-1)$ -tuple of long segments, together with one specific short segment, define a supporting hyperplane corresponding to an unbounded edge of the order- k Voronoi diagram of S . The supporting hyperplane is spanned by an endpoint of each of the d segments. An unbounded edge of the diagram



Figure 2.14. An instance of 5 segments (left), which has one region $\text{reg}(\{s_1, s_2, s_3\})$, shown in blue, on the Gaussian map of the order-3 Voronoi diagram (right) with high complexity.

manifests itself as a vertex in $\text{GM}(\text{VD}_k(S))$. All these vertices are on the boundary of the Gaussian map region of the long segments. \square

We can now prove Theorem 2.2.2.

Proof of Thm. 2.2.2. Let S be a set of n line segments in \mathbb{R}^d . In Theorem 2.3.5, it was stated that there can be $\Omega(k^{d-1})$ vertices in $\text{GM}(\text{VD}_k(S))$ in the worst-case. Each vertex of the Gaussian map corresponds to an edge in $\text{VD}_k(S)$. On the other hand, an edge of the diagram corresponds to at most two vertices in the Gaussian map. Therefore, the diagram contains $\Omega(k^{d-1})$ edges. \square

Theorem 2.3.6. *The complexity of the Gaussian map of the order- k Voronoi diagram of n segments in \mathbb{R}^d is $O(\min\{k, n-k\}n^{d-1})$.*

Proof. We use the point-hyperplane duality transformation T , which establishes a 1-1 correspondence between the upper Gaussian map of the order- k Voronoi diagram and the k th level of the arrangement of d -dimensional wedges. (The lower Gaussian map is constructed in the same manner.) Each segment is mapped to a lower wedge in the dual space, which is bounded by two half-hyperplanes. Let p be a point in dual space. Each wedge below p corresponds to a segment in primal space, which has a non-empty intersection with the open halfspace above $T^{-1}(p)$. Each wedge touching p corresponds to a segment in primal space, which is touching the closed halfspace above $T^{-1}(p)$. Each wedge above p corresponds to a segment in primal space, whose intersection with the closed halfspace above $T(p)$ is empty. Therefore, every point on the k th level of the arrangement of the lower wedges corresponds to a hyperplane in primal space, which supports k segments. The upper or lower envelope of those n wedges, each composed of

two half-hyperplanes, has complexity $O(n^{d-1})$ Edelsbrunner et al. [1989], recalling that we assumed the dimensions d to be constant. Using the bound on the lower envelope, we can now also bound the complexity of the $\leq k$ -level of the arrangement of lower wedges. We apply Theorem 2.1.1 by Clarkson and Shor [1989] to derive a complexity of $O((k+1)^d (\frac{n}{k+1})^{d-1}) = O(kn^{d-1})$. We can derive a similar upper bound of $O((n-k)n^{d-1})$ by using the complexity of the upper envelope of lower wedges as a basis. The upper Gaussian map of the order- k Voronoi diagram corresponds to the k -level of the lower wedges. Combining the two bounds completes the proof. \square

The bounds in Theorems 2.3.5 and 2.3.6 are tight for $n-k = O(1)$. In this case, the complexity of the Gaussian map of VD_k of n segments is $\Theta(n^{d-1})$ in the worst case.

Theorem 2.3.7. *Let S be a set of n line segments in \mathbb{R}^3 . Then, $\text{GM}(\text{FVD}(S))$ can be constructed in worst-case optimal $O(n^2)$ time.*

Proof. We dualize the segments into lower wedges. The upper Gaussian map of the segments corresponds to the upper envelope of the lower wedges in dual space (recall the proof of Thm. 2.3.6). The upper envelope of those wedges, each composed of two halfplanes, is constructed in $O(n^2)$ time Edelsbrunner et al. [1989]. The lower Gaussian map is constructed in the same way. \square

The algorithm of Edelsbrunner et al. [1989] for piecewise-linear functions can be extended to higher dimensions, running in $O(\alpha(n)n^{d-1})$ time Agarwal and Sharir [2000]; Halperin and Sharir [2017]. However, the complexity of the upper envelope of half-hyperplanes is only $O(n^{d-1})$ Edelsbrunner et al. [1989]. We suspect that the same algorithm runs in $O(n^{d-1})$ time when it computes the upper envelope of half-hyperplanes, as in \mathbb{R}^3 , since the complexity of the envelope does not contain the $\alpha(n)$ factor. If so, the Gaussian map of the farthest Voronoi diagram can be constructed in $O(n^{d-1})$ time.

2.4 Lines as Sites

Let S be a set of lines in \mathbb{R}^d . We assume that the lines are in general position, *i.e.*, the lines are non-intersecting and the directions of any d lines are linearly independent. In this section we derive similar conditions for the order- k Voronoi diagram of lines to have unbounded cells in some direction.

Definition 2.4.1. *For a line ℓ and a direction \vec{v} , the angular distance $\angle(\vec{v}, \ell)$ is the smallest angle between \vec{v} and the direction of ℓ , see Fig. 2.15a.*

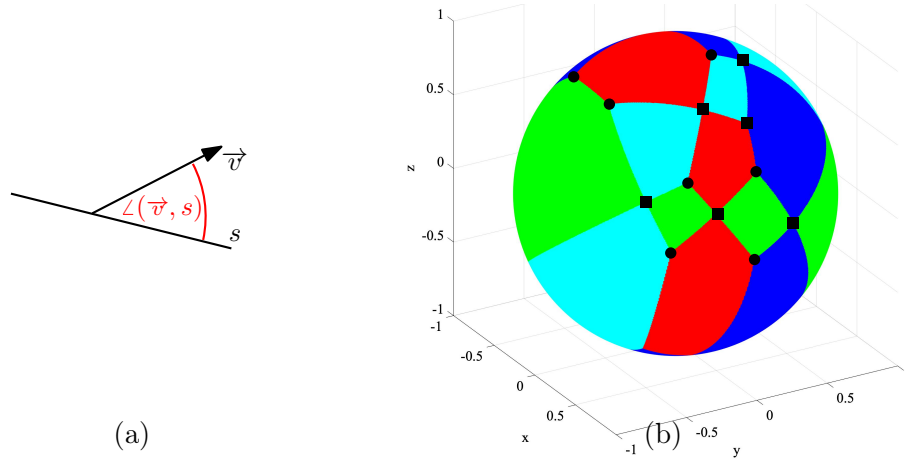


Figure 2.15. (a) The angular distance $\angle(v, \ell)$ between line ℓ and direction v . (b) GM(FVD) of four lines in \mathbb{R}^3 . The farthest regions of the lines are colored in different colors. Vertices of anomaly are shown with squared boxes; proper vertices with disks.

Definition 2.4.2. Let L be a set of lines, and let H be a subset of L . An angle β is a supporting angle of H in direction \vec{v} if

1. The angular distance between \vec{v} and any of the lines in H is at most β ; and
2. The angular distance between \vec{v} and any of the lines in $L \setminus H$ is at least β , and at least one site in $L \setminus H$ realizes the angular distance β .

Theorem 2.4.3. A set of lines H , with $|H| = k$, induces an unbounded region in direction \vec{v} in $VD_k(L)$ if and only if there exists a supporting angle of H in direction \vec{v} .

Proof. Let ℓ be a line, and r be a ray emanating from an arbitrary point p in direction \vec{v} . With some calculations, one can derive that

$$\lim_{\lambda \rightarrow \infty} \frac{d(\ell, p + \lambda \cdot \vec{v})}{\lambda} = \sin(\angle(\vec{v}, \ell)).$$

The distance between the line and a point on the ray r increases with rate $\sin(\angle(\vec{v}, \ell))$ as the point moves along the ray to infinity. Therefore, the set of k lines, which minimizes the angular distance to \vec{v} , has an unbounded cell in direction \vec{v} . \square

Corollary 2.4.4. A supporting angle of H , which is realized by i lines (at least one of which is in H), corresponds to an unbounded $(d-i+1)$ -cell in the order- k Voronoi diagram of L .

All d -cells, which are unbounded in the same direction \vec{v} , touch at a common lower-dimensional cell. This cell is determined by the lines, which have the same angular distance to \vec{v} . A cell, which is equidistant to i lines, is $(d-i+1)$ -dimensional.

Theorem 2.4.5. *A supporting angle β of H in direction \vec{v} , which is realized by i lines (of which, at least one belongs to H), corresponds to a $(d-i)$ -cell (resp., $(d-i-1)$ -cell) in $\text{GM}(\text{VD}_k(L))$, if $\beta < \pi/2$ (resp., $\beta = \pi/2$).*

Proof. Assume first that $\beta < \pi/2$. The sites are split into three types: lines which have smaller, equal, or larger angular distance to \vec{v} than the supporting angle. How can we rotate the direction \vec{v} while still keeping the same partitioning of sites? In particular, having the same set of sites with the same angular distance to \vec{v} as the supporting angle? If only one line had the same angular distance to \vec{v} , then the direction \vec{v} could be rotated freely. Each additional line with the same angular distance introduces one constraint on the rotation.

Now assume that $\beta = \pi/2$. Again, we consider the partition of the lines with smaller or equal angular distance to \vec{v} than $\pi/2$. If a line has angular distance $\pi/2$ to the direction \vec{v} , then this angular distance can be measured in both unbounded directions of the line. How can we rotate the direction \vec{v} while still keeping the same partitioning of sites? In particular, having the same set of sites with angular distance $\pi/2$? If just one line had the same angular distance, then the direction \vec{v} could be rotated in $d-2$ dimensions. Each additional line with the same angular distance introduces one more constraint on the rotation. \square

Typically, i -cells of the Gaussian map correspond to $(i+1)$ -cells of the corresponding Voronoi diagram. The only exceptions are cells whose supporting angle is $\pi/2$, which correspond to $(i+2)$ -cells of VD_k .

Definition 2.4.6. *The i -cells of the Gaussian map, $i < d - 1$, which correspond to a supporting angle of $\pi/2$, are called cells of anomaly. All other cells are called proper.*

In \mathbb{R}^3 , the only cells of anomaly are vertices, see Fig. 2.15b. Such a vertex corresponds to a direction in which the bisector of two lines seems to be self-intersecting. The bisector of two lines ℓ, ℓ' is a hyperbolic paraboloid. Seen "from infinity" this hyperbolic paraboloid looks like two intersecting planes. The intersection of those planes is a line l , which is unbounded in two antipodal directions $-\vec{v}, \vec{v}$, which are the vertices of anomaly on the Gaussian map. One of the lines ℓ, ℓ' is actually strictly closer to direction \vec{v} than the other. Only "at infinity," both lines seem to have equal distance in direction \vec{v} .

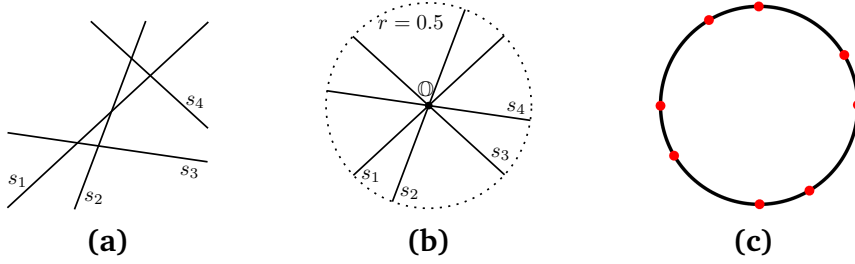


Figure 2.16. (a) Lines L and their (b) transformed segments $\tau(L)$ have identical (c) Gaussian maps $\text{GM}(\text{VD}_2(L)) = \text{GM}(\text{VD}_2(\tau(L)))$.

In general space \mathbb{R}^d , the i -cells of anomaly on the Gaussian map correspond to $(i+2)$ -cells in the order- k Voronoi diagram. Looking at the Gaussian map, these $(i+2)$ -cells seem as if they intersect, however, they do not intersect in the actual diagram. Let \vec{v} be the direction of a cell of anomaly. The lines, which are orthogonal to \vec{v} , can actually be ordered along direction \vec{v} . Let j be the number of lines that are not orthogonal to \vec{v} . The region of those j lines, together with the closest $k-j$ orthogonal lines, is unbounded in direction \vec{v} and, moreover, is not split by an $(i+1)$ -cell in direction \vec{v} .

We define a transformation τ that maps lines to segments. Each line ℓ is mapped to a unit segment $\tau(\ell)$ that has the same direction as the line and the origin O as midpoint, see Fig. 2.16. When applied to a set of lines, the result of the transformation is a set of segments in non-general position, but this does not affect the upper bound on the complexity of the Gaussian map.

Theorem 2.4.7. *Let L be a set of lines. Then, $\text{GM}(\text{VD}_k(L)) = \text{GM}(\text{VD}_k(\tau(L)))$.*

Proof. A set of segments has an unbounded cell in direction \vec{v} if and only if there exists a supporting hyperplane for those segments in direction \vec{v} . The supporting hyperplane P separates k segments $H \subset \tau(L)$, which have non-empty intersections with the closure of halfspace P^+ , from the other segments. The properties that the segments have unit length and the origin as midpoint guarantee that all segments in H have a smaller angular distance to \vec{v} than any of the other segments in $\tau(L) \setminus H$. \square

As a consequence, lower bounds on the worst-case complexity of the Gaussian map, derived for lines as sites, carry over to segments as sites. In the same manner, all upper bounds on the worst-case complexity on the Gaussian map for segments also apply to lines. In addition, the algorithm of Theorem 2.3.7 to construct the Gaussian map of the farthest Voronoi diagram extends to lines as sites. (Note that the algorithm does not require the segments to be in general position.)

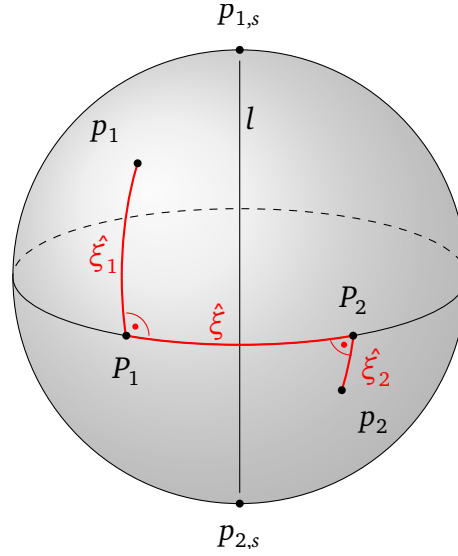


Figure 2.17. Construction of the path $(\hat{\xi}_1, \hat{\xi}, \hat{\xi}_2)$ for lines as sites.

Corollary 2.4.8. *The Gaussian map of the order- k Voronoi diagram of n lines in \mathbb{R}^d has $O(\min\{k, n-k\}n^{d-1})$ complexity. The Gaussian map can be constructed in $O(n^{d-1}\alpha(n))$ time, while if $d = 3$, the time drops to $O(n^2)$.*

Theorem 2.4.9. *Let L be a set of lines. Then, $\text{FVD}(L)$ does not have tunnels.*

For this theorem to make sense, we need to clarify our interpretation of tunnels in the presence of cells of anomaly. Recall that cells of anomaly do not correspond to any cell in the Voronoi diagram. For defining the Gaussian map, we compute the intersection of the unit sphere with the Voronoi diagram scaled by a factor λ . The Gaussian map is then derived when λ reaches 0 in the limit. We say that a cell c of the Voronoi diagram forms a tunnel, if the intersection of the unit sphere with the scaled cell c is disconnected, for every arbitrarily small (but positive) scaling factor λ . Effectively, cells of anomaly can connect some $(d-1)$ -cells of the Gaussian map, while they disconnect others.

Proof. Let p_1 and p_2 be two points on $\text{GM}(\text{FVD}(L))$ in a farthest cell of line ℓ . The line ℓ is unbounded in two directions corresponding to two points in $\text{GM}(\text{FVD}(L))$, which we call $p_{1,\ell}$ and $p_{2,\ell}$, see Fig. 2.17. We define the path $\hat{\xi}_1 \subset \text{GM}(\text{FVD}(L))$ by starting from p_1 and moving along the geodesic away from the closer point of $p_{1,\ell}$ and $p_{2,\ell}$, until both points $p_{1,\ell}$ and $p_{2,\ell}$ have equal distance to our current point on $\hat{\xi}_1$. We call the last point of this path P_1 . Similarly, we define the path $\hat{\xi}_2$

and point P_2 in the cell ϕ_2 . Note that by construction, the entire paths $\hat{\xi}_1$ and $\hat{\xi}_2$ contain only points corresponding to ℓ as the farthest line.

Now we perform the same procedure as in the proof of Theorem 2.3.4. The two points P_1 and P_2 represent directions \vec{v}_1 and \vec{v}_2 , along which there exist points $x_1, x_2 \in \text{freg}(\ell)$ for which $\vec{v}_i = \overrightarrow{q_i x_i}$ (with $i = 1, 2$), where q_i is the point on ℓ realizing the distance between x_i and ℓ . Since x_1 and x_2 are contained in the same cell of $\text{freg}(\ell)$, there exists a continuous path ξ connecting both points and being fully contained in this cell. We can map every point $x \in \xi$ to the direction $\vec{v} = \overrightarrow{q x}$, with q realizing the distance between x and ℓ , which we afterwards map to its corresponding point $p \in \text{GM}(\text{FVD}(L))$. Note that the point p is contained in a farthest cell corresponding to line ℓ . By continuity, mapping the whole path ξ to $\text{GM}(\text{FVD}(L))$ yields a continuous path $\hat{\xi}$ between P_1 and P_2 . The path $(\hat{\xi}_1, \hat{\xi}, \hat{\xi}_2)$ consisting solely of directions in which line ℓ is the farthest site. \square

A similar construction, as in Section 2.3, can be used for showing that $\text{VD}_k(L)$ can have tunnels for a set of lines L and $k \leq n - d$.

The following result stands by its own and will be used to analyze the number of d -cells in the farthest Voronoi diagram of lines and its Gaussian map. We look at an arrangement of great spheres with the same center and radius on a $(d-1)$ -sphere. For example, consider the 2-dimensional unit sphere \mathbb{S}^2 in \mathbb{R}^3 and n great circles on it. We answer the following question: “Into how many 2-dimensional faces is the unit sphere split by the great circles?” We assume that no d great spheres have a point in common.

Theorem 2.4.10. *Let \mathbb{S} be a set of n many $(d-2)$ -dimensional unit hyperspheres in \mathbb{R}^d , centered at the origin. Then, the arrangement of \mathbb{S} on the $(d-1)$ -dimensional unit hypersphere \mathbb{S}^{d-1} contains $\binom{n-1}{d-1} + \sum_{k=0}^{d-1} \binom{n}{k}$ many $(d-1)$ -cells.*

Proof. Each hypersphere $s \in \mathbb{S}$ is the intersection of \mathbb{S}^{d-1} with a hyperplane through the origin. Therefore, the arrangement \mathbb{S} on \mathbb{S}^{d-1} can be seen as an arrangement of hyperplanes in \mathbb{R}^d , each containing the origin \mathbb{O} , intersected with \mathbb{S}^{d-1} . In order to compute the combinatorial structure of this intersection, we employ a bijection—a gnomonic projection of the lower and upper hemispheres of \mathbb{S}^{d-1} (except its equator) to the two hyperplanes tangent to the poles of \mathbb{S}^{d-1} ($x_d = -1$ and $x_d = 1$), respectively. We map every point p on the lower hemisphere to the intersection of the line $\overline{\mathbb{O}p}$ with the plane $x_d = -1$ and, similarly, every point on the upper hemisphere to the intersection of the line $\overline{\mathbb{O}p}$ with the plane $x_d = 1$. This mapping maps the $(d-2)$ -dimensional spheres in \mathbb{S} to $(d-2)$ -dimensional hyperplanes on the hyperplanes $x_d = -1$ and $x_d = 1$. See Figure 2.18 for an

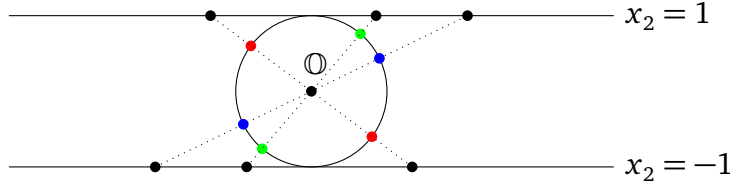


Figure 2.18. Example for $d = 2$ and $n = 3$: Three 0-dimensional unit spheres (blue, green, red) split the unit circle into 6 arcs.

example with $d = 2$ and $n = 3$. Such hyperplane arrangements have $\sum_{k=0}^{d-1} \binom{n}{k}$ many $(d-1)$ -dimensional parts, among which $\binom{n-1}{d-1}$ are bounded Stanley [2004]. The number of features on the hypersphere \mathbb{S}^{d-1} is, hence, the sum of features on both hyperplane arrangements minus the number of unbounded features of one hyperplane arrangement. The unbounded features on the hyperplane arrangements are glued together on the hypersphere and, therefore, only have to be counted once. Counting the $(d-1)$ -dimensional parts, we obtain the claimed count. \square

Theorem 2.4.11. *Let L be a set of n lines. The Gaussian map of $\text{FVD}(L)$ has $\Theta(n^{d-1})$ many $(d-1)$ -cells.*

Proof. We consider, for each line, the orthogonal directions. We get n many $(d-2)$ -dimensional hyperspheres in total. Each of those hyperspheres is partitioned into $\binom{n-2}{d-2} + \sum_{k=0}^{d-2} \binom{n-1}{k}$ parts by the other $(n-1)$ -hyperspheres due to Theorem 2.4.10. A direction in one of those parts is orthogonal to exactly one line in L and, hence, is also part of the farthest Voronoi region of that line. In total, all n hyperspheres are split into $n \left(\binom{n-2}{d-2} + \sum_{k=0}^{d-2} \binom{n-1}{k} \right) = \Theta(n^{d-1})$ parts. Now, consider a direction \vec{v} not on any hypersphere but in the farthest region of ℓ . The shortest path on the Gaussian map from \vec{v} to the hypersphere corresponding to line ℓ contains only directions of $\text{freg}(\ell)$. Therefore, there are no additional $(d-1)$ -cells not containing a part of a hypersphere. \square

It is easy to prove that all cells of $\text{GM}(\text{FVD}(L))$ are convex, in the sense that the shortest path between any two points of a cell is contained in that cell.

For a set of lines L in \mathbb{R}^3 , we count the number of 2-cells of $\text{GM}(\text{FVD}(L))$ and subtract the number of vertices of anomaly to derive the exact number of 3-cells in $\text{FVD}(L)$.

Theorem 2.4.12. *Let L be any set of $n \geq 2$ lines in \mathbb{R}^3 . Then, $\text{FVD}(L)$ has exactly $n^2 - n$ many 3-cells.*

An unbounded i -cell of a cell complex M may correspond to many $(i-1)$ -cells in the Gaussian map of M . Therefore, we need to study carefully the Gaussian map in order to derive a lower bound on the complexity of M .

Proof. We know from the proof of Theorem 2.4.11 that $\text{GM}(\text{FVD}(L))$ in \mathbb{R}^3 has

$$n \left(\binom{n-2}{1} + \sum_{k=0}^1 \binom{n-1}{k} \right) = 2n^2 - 2n$$

2-cells. Theorem 2.4.9 states that each of those 2-cells of the Gaussian map corresponds to a 3-cell of the farthest Voronoi diagram, unless the 2-cells are connected through vertices of anomaly. Each vertex of anomaly is the meeting point of exactly four 2-cells of the Gaussian map. In the actual $\text{FVD}(L)$, two of these four 2-cells correspond to the same 3-cell. Therefore, we need to subtract the number of vertices of anomaly from the number of 2-cells on the Gaussian map to derive the number of 3-cells of the farthest Voronoi diagram. There are two vertices of anomaly for every pair of lines. Hence, the farthest Voronoi diagram has $2n^2 - 2n - 2 \cdot \binom{n}{2} = n^2 - n$ many 3-cells. \square

Theorem 2.4.13. *The worst-case complexity of FVD of n lines is $\Omega(n^{d-1})$.*

Proof. We bound the number of proper vertices (not those of anomaly) of $\text{GM}(\text{FVD}(L))$ from below. Those vertices correspond to unbounded edges of the farthest Voronoi diagram. The set of orthogonal directions to a line is a hypersphere of dimension $d-2$ in $\text{GM}(\text{FVD}(L))$. By Theorem 2.4.11, the hyperspheres of all lines partition $\text{GM}(\text{FVD}(L))$ into $\Theta(n^{d-1})$ many $(d-1)$ -dimensional parts. If $n \geq d$ (which is the case in the asymptotic analysis), each of those parts contains at least one proper vertex. Then, $\text{FVD}(L)$ has an unbounded edge in that direction. Each edge is unbounded in at most two directions. Hence, the number of edges can be bounded from below by half of the number of proper vertices of the Gaussian map. Thus, the number of edges in $\text{FVD}(L)$ is $\Omega(n^{d-1})$. \square

2.5 Combination of Lines and Segments

Let $E = L \cup S$ be a combination of $|L| = m$ many lines and $|S| = n - m$ many line segments in \mathbb{R}^d . We again assume that the lines are in general position, *i.e.*, the lines are non-intersecting and the directions of any d lines are linearly independent. Moreover, we assume that there is no hyperplane that contains more than d endpoints of sites, where lines contained in such a hyperplane are counted twice (as if the line were a segment) and otherwise not at all. In this

section, we describe the structure of the Gaussian map of the order- k Voronoi diagram of E , and also show how it can be computed. Using the same proof technique as Theorem 2.4.9, the “no-tunnel” property holds for a combination of lines and line segments as sites, *i.e.*, a path from p to q within a cell c of the Voronoi diagram can be mapped to the Gaussian map.

Observation 2.5.1. *Let E be a set of lines and segments. Then, $\text{FVD}(E)$ does not have tunnels.*

As already mentioned before in the proof of Theorem 2.4.3, the distance of a line to a point, which moves along a ray to infinity, increases at a rate less than 1, if the line and ray are not orthogonal. On the other hand, the distance of a segment to such a moving point always increases at a rate of 1 in the limit. Intuitively speaking, looking at the sites from arbitrarily far along a fixed direction all the segments are further away than the lines, which are not orthogonal to that direction. Only lines that are orthogonal to the direction of consideration play the same role as segments.

Theorem 2.5.2. *Let E be a set of lines and segments, and let $H \subseteq E$ be of cardinality k . The set of sites H induces an unbounded cell in direction \vec{v} in $\text{VD}_k(E)$ if and only if either one of the following condition holds.*

1. *There are less than k lines non-orthogonal to \vec{v} and there exists a supporting hyperplane in direction \vec{v} of sites H ; or*
2. *There are at least k lines non-orthogonal to \vec{v} , the set H consists of lines, and there exists a supporting angle in direction \vec{v} of sites H .*

The proof of Theorem 2.5.2 proceeds analogously to the proof of Theorem 2.3.2 and is the reason for the general position assumption on the sites described in the beginning of this section.

Definition 2.5.3. *For a line $\ell \in L$, we define the great sphere G_ℓ to be the set of directions, which are orthogonal to ℓ . Moreover, for $L' \subset L$ define $D_{L'}^- := \bigcap_{\ell' \in L'} G_{\ell'} \setminus \bigcup_{\ell \in L \setminus L'} G_\ell$ to be the set of directions, which are orthogonal to exactly all lines in L' but no other. Similarly let $D_{L'}^+ := \bigcap_{\ell' \in L'} G_{\ell'}$ be the set of directions, which are orthogonal to at least all lines in L' . If $L' = \emptyset$, then we let $D_{L'}^-$ and $D_{L'}^+$ be the set of all possible directions \mathbb{S}^{d-1} .*

Observe that due to our general position assumption for the lines, the set $D_{L'}^-$ is empty for $|L'| \geq d$. Moreover, the sets $D_{L'}^-$ for $L' \subset L$ and $|L'| < d$ form a partition of the sphere of directions \mathbb{S}^{d-1} .

We will now describe the structure of $\text{GM}(\text{VD}_k(E))$ within each such domain $D_{L'}^-$. For any direction $\vec{v} \in D_{L'}^-$, there are exactly $m - |L'|$ many lines non-orthogonal to it. Therefore, we derive the following relation according to Theorem 2.5.2.

$$\text{GM}(\text{VD}_k(E))|_{D_{L'}^-} = \begin{cases} \text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^-} & \text{if } k > m - |L'| \\ \text{GM}(\text{VD}_k(L))|_{D_{L'}^-} & \text{if } k \leq m - |L'| \end{cases} \quad (2.4)$$

Note that in the first case we slightly abuse notation in the following sense. The regions of $\text{GM}(\text{VD}_k(E))|_{D_{L'}^-}$ are defined by tuples of k many sites, whereas the regions of $\text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^-}$ belong to sets of sites of size $k - m$. The difference is caused by the lines $L \setminus L'$. With the equality symbol we want to express that the cell complexes of the Gaussian maps are the same.

We will now characterize the right-hand side of Eq. (2.4). More precisely we want to bound its complexity and also give an algorithm to construct it even in a slightly bigger domain, namely $D_{L'}^+$. The domain $D_{L'}^+$ is a great sphere of dimension $d - |L'| - 1$ and we will show that the Gaussian map in d dimensions restricted to $D_{L'}^+$ actually corresponds to a Gaussian map of sites in $d - |L'|$ dimensions. Let us formalize this intuition in the next lemmas.

Lemma 2.5.4. *Let E be a set of lines and line segments, and let H be a subset of E . For a linear subspace Q , we denote by Proj_Q the orthogonal projection to Q . Let P be a hyperplane orthogonal to Q . Then, P is a supporting hyperplane of sites H if and only if $\text{Proj}_Q(P)$ is a supporting hyperplane of sites $\text{Proj}_Q(H)$.*

Proof. The four properties for being a supporting hyperplane in direction \vec{v} are invariant under orthogonal projection to a hyperplane which contains direction \vec{v} . \square

Let Q be a linear subspace and D its set of directions it contains. Essentially Lemma 2.5.4 implies, that for an arbitrary set of sites E it holds that $\text{GM}(\text{VD}_k(E))|_D = \text{GM}(\text{VD}_k(\text{Proj}_Q(E)))$. Most importantly, the projection Proj_Q of all sites can be constructed in $O(n)$ time, and the Gaussian map in lower dimensional space can be constructed faster.

Lemma 2.5.5. *Let $L' \subset L$, then $\text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^+}$ has $O(\min\{k, n-k\}n^{d-|L'|-1})$ complexity. Further, if $k = n - 1$ and $d - |L'| = 3$, it can be constructed in $O(n^2)$ time.*

Proof. Let Q be the linear subspace, which contains exactly all directions of $D_{L'}^+$. We first project all sites $S \cup L'$ orthogonally to Q . Note that all lines L' are

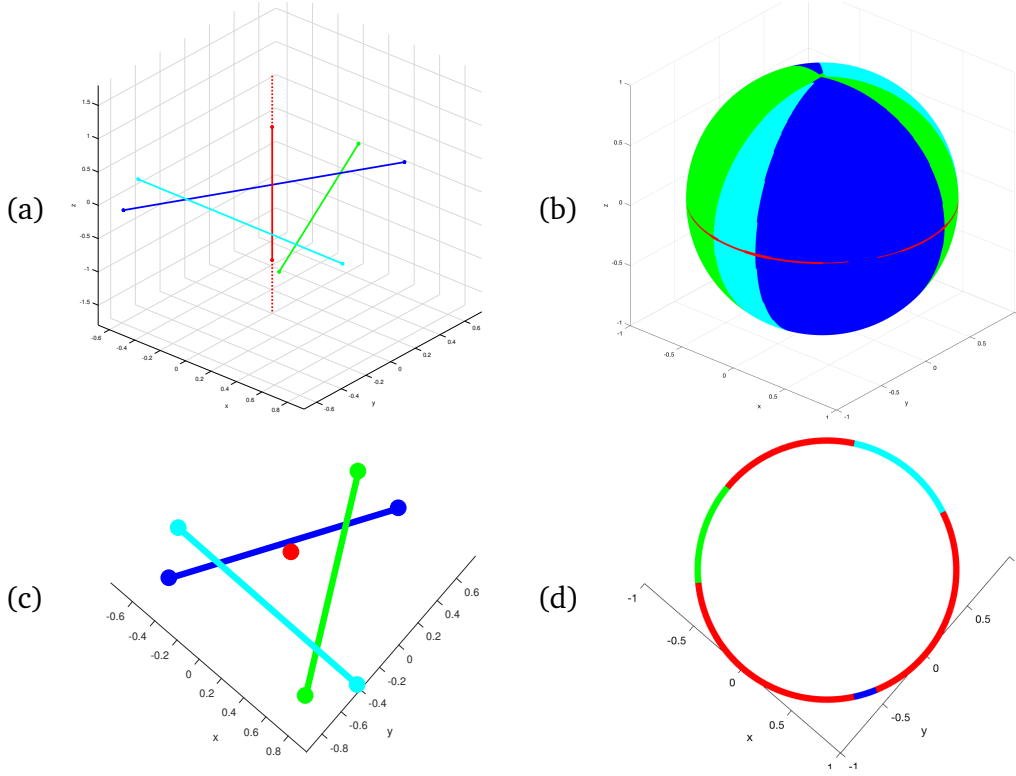


Figure 2.19. Illustration of Lemma 2.5.4. (a) The sites E (1 vertical red line and 3 line segments) in 3-space; (b) $\text{GM}(\text{FVD}(E))$ in \mathbb{R}^3 ; (c) The sites E projected to the x - y -plane Q (d) $\text{GM}(\text{FVD}(\text{Proj}_Q(E)))$ in \mathbb{R}^2 , which corresponds to the equator of $\text{GM}(\text{FVD}(E))$.

orthogonal to Q and therefore they are projected to points. Hence $\text{Proj}_Q(S \cup L')$ is a set of segments, some of which degenerate to points. The space Q has dimension $d - |L'|$ and consequently $\text{GM}(\text{FVD}(\text{Proj}_Q(S \cup L')))$ has $O(\min\{k, n-k\}n^{d-|L'|-1})$ complexity by Theorem 2.3.6. If $k = n-1$ and $d - |L'| = 3$, then $\text{GM}(\text{FVD}(\text{Proj}_Q(S \cup L')))$ can be constructed in $O(n^2)$ time, see Theorem 2.3.7. Due to Lemma 2.5.4, this Gaussian map is the same as $\text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^+}$. \square

So far we have characterized the pieces of $\text{GM}(\text{VD}_k(E))$ and it remains to put them together. This merging idea is formulated in Algorithm 1. We also use this construction to argue about the Gaussian map's combinatorial complexity.

Theorem 2.5.6. *The complexity of the Gaussian map of the order- k Voronoi diagram of a combination of n lines and segments in \mathbb{R}^d is $O(\min\{k, n-k\}n^{d-1})$.*

Proof. Algorithm 1 gives the outline for our complexity analysis. Depending on the relation between k and m , the initial map G in \mathbb{R}^d is either a Gaussian map of

Algorithm 1: Building $\text{GM}(\text{VD}_k(S \cup L))$

```

1 if  $k > m$  then
2   |  $G \leftarrow \text{GM}(\text{VD}_{k-m}(S));$ 
3 else
4   |  $G \leftarrow \text{GM}(\text{VD}_k(L));$ 
5 for  $i = 1 \rightarrow d - 1$  do
6   | if  $k > m - i$  then
7     | | for all  $L' \subset L$  with  $|L'| = i$  do
8       | | |  $G|_{D_{L'}^+} \leftarrow \text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^+};$ 
9 return  $G;$ 

```

an order- $(k - m)$ diagram of $(n - m)$ many sites or a Gaussian map of an order- k Voronoi diagram of m sites. In both cases the complexity can be bounded by $O(\min\{k, n-k\}n^{d-1})$ due to Theorem 2.3.6. After the initialization the map G is the correct Gaussian map in all directions, which are not orthogonal to any lines. It remains to update the map G along the great spheres and bound the additional complexity in line 9 of Algorithm 1. Let us now consider the i -th update step of the for loop in Line 6. We need to consider all tuples of lines of size i , of which there are $O(d^i)$ many. For each such set of lines L' , we update G with a Gaussian map $\text{GM}(\text{VD}_{k-m+|L'|}(S \cup L'))|_{D_{L'}^+}$, which according to Lemma 2.5.5 has $O(\min\{k, n-k\}n^{d-i-1})$ complexity. Updating for all such tuples of size i therefore increases the total complexity by $O(\min\{k, n-k\}n^{d-1})$. Since i runs between 1 and $d - 1$ the total complexity of G hence does not increase asymptotically, recalling that d is assumed to be constant. Finally we have to address one detail, that was omitted so far. When we update G along a great sphere, then this sphere may actually cut features of the map in two pieces, thus also increasing the complexity of the map. Let us go back to the time when G was updated for all tuples of lines of size $i - 1$. For a fixed set of lines L' of size i , the great sphere $D_{L'}^+$ splits some features of G . Consider a direction \vec{v} very close to $D_{L'}^+$. The lines in L' may belong to the closest k sites in direction \vec{v} but might not do so on $D_{L'}^+$. Essentially, the number of features that are split by updating along $D_{L'}^+$ can be bounded by the complexity of $\text{GM}(\text{VD}_{k-m}(S))|_{D_{L'}^+}$, whose complexity can also be bounded from above by $O(\min\{k, n-k\}n^{d-i-1})$. Thus even the split features do not asymptotically increase the overall complexity. \square

Let us now look into actually constructing the Gaussian map of the farthest Voronoi diagram of lines and line segments combined in three dimensional space.

But first we need to introduce the concept of a zone of great circles on a sphere similarly to the zone of a line in an arrangement of lines.

Definition 2.5.7. Let \mathbb{S} be a set of n many $(d-2)$ -dimensional unit hyperspheres in \mathbb{R}^d , centered at the origin. These n hyperspheres form an arrangement \mathcal{A} on the $(d-1)$ -dimensional unit hypersphere \mathbb{S}^{d-1} . The zone of one $(d-2)$ -dimensional hypersphere s in \mathcal{A} is defined as the set of cells of \mathcal{A} , which bound the $(d-1)$ cells intersected by s .

Theorem 2.5.8 (Zone theorem for great spheres). *The complexity of the zone within an arrangement of hyperspheres, as defined in Definition 2.5.7, is $O(n^{d-2})$.*

Proof. Similarly to Theorem 2.4.10, we employ the same gnomonic projection of the lower and upper hemispheres of \mathbb{S}^{d-1} (except its equator) to the two hyperplanes tangent to the poles of \mathbb{S}^{d-1} ($x_d = -1$ and $x_d = 1$), respectively. This projection maps the arrangement of n many $(d-2)$ -dimensional hyperspheres in \mathbb{R}^d to two arrangements of hyperplanes in \mathbb{R}^{d-1} . According to Edelsbrunner et al. [1993], the complexity of a zone in the arrangement of n hyperplanes in \mathbb{R}^d is $O(n^{d-1})$. The zone within an arrangement of hyperspheres corresponds to the union of the zones in the arrangement of hyperplanes. Therefore we can also bound the complexity of the zone of hyperspheres by $O(n^{d-2})$. \square

Theorem 2.5.9. *Let E be a combination of n lines and segments in \mathbb{R}^3 . Then $\text{GM}(\text{FVD}(E))$ can be constructed in worst-case optimal $O(n^2)$ time.*

Proof. We use Algorithm 1 to construct the Gaussian map. The initial map of Line 2 or 4 can be constructed in $O(n^2)$ time according to Theorem 2.3.7. Afterwards the map G is updated along great circles and their intersections. We need to explain the following two key steps:

- the Gaussian map along the great circles and their intersections can be computed efficiently enough and
- merging the original map G with the updates along the great circles.

Regarding the first point, note that computing a Gaussian map of a farthest Voronoi diagram of segments can be computed from scratch in $O(n \log n)$ time, see Aurenhammer et al. [2006]. Unfortunately in our case, this would not be efficient enough, because we would have to compute m such maps, one for each line, resulting in an $O(n^2 \log n)$ running time. In order to be more efficient, we take advantage of the already initially computed map G . Let $\ell \in L$ be fixed. The map G along the D_ℓ^- corresponds to the Gaussian map of the sites $E \setminus \{\ell\}$. In

the update step we need to update the map such that it includes the site ℓ . As described in Theorem 2.3.7, the Gaussian map corresponds to the upper envelope of lower wedges in the plane. This upper envelope has $O(n)$ complexity and adding one wedge, corresponding to line ℓ , can be easily done in linear time. Hence computing the $\text{GM}(\text{FVD}(E))$ along all great circles defined by L takes in total $O(n^2)$ time. Finally we need to look at the $O(n^2)$ many directions, which are orthogonal to pairs of lines. Let $\ell, \ell' \in L$ and \vec{v} a direction orthogonal to both of the lines. The farthest site in direction \vec{v} can be simply determined in constant time by comparing the farthest site of the initial map G in direction \vec{v} with the two lines ℓ and ℓ' . It remains to merge all the pieces of the Gaussian map. Similarly to adding a line to an arrangement of lines in the plane, we can add great circles to an arrangement of great circles on a sphere. Due to the Zone theorem 2.5.8 adding one such great circle takes $O(n)$ time, hence taking $O(n^2)$ time in total for updating the Gaussian map along all great circles. The complexity of the Gaussian map along each great circle is $O(n)$ and therefore traversing the great circle and updating the directions which are orthogonal to two lines can also be done in linear time for each great circle, thus not changing the asymptotic running time. \square

2.6 Polyhedra or Clusters of Points as Sites

Let E be a set of bounded polyhedra. We assume that the sites are in general position, *i.e.*, no $d+1$ many vertices lie on the same hyperplane.

Theorem 2.3.2 directly generalizes for this setting.

Theorem 2.6.1. *A set of convex polyhedra H , with $|H| = k$, induces an unbounded region in direction \vec{v} in the order- k Voronoi diagram of polyhedra E , if and only if there exists a supporting hyperplane of H in direction \vec{v} .*

Note that only extreme points of the convex hull of the site determine the site's unbounded directions.

Theorem 2.6.2. *The complexity of the Gaussian map of the order- k Voronoi diagram of bounded polyhedra with n vertices in total in \mathbb{R}^3 is $O(\min(k^2n, (n-k)n^2\alpha(n)))$.*

Proof. The proof idea is the same as in Theorem 2.3.6. We use the point-hyperplane duality transformation T , which establishes a 1-1 correspondence between the upper Gaussian map of the order- k Voronoi diagram of the polyhedra and the k -th level of an arrangement of polyhedra. (The lower Gaussian map is constructed in the same manner.) Each vertex of a site is mapped to a

hyperplane in the dual space. The site corresponds to the lower envelope \mathcal{P} of these hyperplanes in the dual space. If a site consists of m points, then \mathcal{P} is an unbounded convex polyhedron with at most m facets, which can be decomposed into $O(m)$ many triangles.

The upper envelope of those polyhedra in dual space has complexity $O(n^2\alpha(n))$ Edelsbrunner et al. [1989]. The lower envelope of the polyhedra in dual space has complexity $O(n)$ McMullen [1970], as it is essentially the lower envelope of $O(n)$ hyperplanes.

Applying Theorem 2.1.1 by Clarkson and Shor [1989], we derive upper bounds on the complexity of the $\leq k$ -level of the arrangement of unbounded polyhedra:

$$O\left((k+1)^3 \binom{n}{k+1}\right) = O(k^2n),$$

which follows from the lower envelope, and

$$O\left((n-k)^3 \left(\frac{n}{n-k}\right)^2 \alpha\left(\frac{n}{n-k}\right)\right) = O((n-k)n^2\alpha(n/(n-k))),$$

which follows from the upper envelope. The upper Gaussian map of the order- k Voronoi diagram corresponds to the k -level of the unbounded polyhedra. Combining the two bounds completes the proof. \square

Note that the same approach would also work to bound the complexity of the Gaussian map of the order- k Voronoi diagram in $d \geq 3$ dimensions, but unfortunately the bounds would become much worse. The main reason is that the decomposition of the lower envelopes \mathcal{P} into simplices can have $\omega(m)$ simplices in higher dimensions.

Theorem 2.6.3. *Let E be a set of bounded polyhedra with n vertices in total in \mathbb{R}^3 . Then, $\text{GM}(\text{FVD}(E))$ can be constructed in $O(n^2\alpha(n))$ time.*

Proof. Recall the proof of Theorem 2.6.2. The lower envelope \mathcal{P} of each site can be constructed in $O(n^2)$ total time Chazelle [1991]. Decomposing each dual polyhedron \mathcal{P} into triangles can be done in time linear in its number of faces. Computing the upper envelope of these polyhedra, which is composed of $O(n)$ many triangles, takes $O(n^2\alpha(n))$ time Edelsbrunner et al. [1989]. This envelope corresponds to the upper Gaussian map of the sites. The lower Gaussian map is constructed in the same way. \square

Let a set of points in \mathbb{R}^d be called a *cluster*, and let E be a family of m such clusters of points in \mathbb{R}^d , where the total number of points is n . We assume that

the sites are in general position, *i.e.*, no $(d+1)$ points lie on the same hyperplane. The distance from a point $x \in \mathbb{R}^d$ to a site $e \in E$ can be measured in 2 different ways, depending if the closest or farthest point in a cluster is considered.

- $d_{\min}(x, e) = \min\{d(x, y) | y \in E\}$
- $d_{\max}(x, e) = \max\{d(x, y) | y \in E\}$

We can define two different types of *cluster Voronoi diagrams* on E depending on the distance function under consideration, d_{\min} or d_{\max} . The nearest Voronoi diagram induced by d_{\max} on E has been termed the Hausdorff Voronoi diagram of E and has been essentially considered in the plane, see e.g., Edelsbrunner et al. [1989]; Papadopoulou and Lee [2004]. On the other hand, the farthest Voronoi diagram of E using d_{\min} was first considered by Huttenlocher et al. [1993], and it has also been termed the *farthest color Voronoi diagram*, see e.g., Mantas et al. [2021] and references therein.

The nearest (resp., farthest) Voronoi diagram induced by d_{\max} (resp., d_{\min}) corresponds to the Hausdorff (resp., farthest cluster) Voronoi diagram Mantas et al. [2021]; Papadopoulou [2004]. For $H \subset E$, denote by $\text{reg}_{k-\max}(H)$ (resp., $\text{reg}_{k-\min}(H)$) the Voronoi regions of H induced by d_{\min} (resp., d_{\max}).

Theorem 2.6.4. *The Gaussian map of the order- k Voronoi diagram of clusters as sites induced by d_{\min} is the same as the Gaussian map of the order- k Voronoi diagram of bounded polyhedra, if the clusters consist of the vertices of the polyhedra.*

Proof. The distance to a bounded polyhedron, seen from a point at “infinity,” is realized by a vertex of the polyhedron. \square

It was pointed out Papadopoulou [2004]; Mantas et al. [2021] that there is a strong relation between the unbounded directions of the farthest cluster Voronoi diagram and the Hausdorff diagram.

Observation 2.6.5. *Let E be a set of clusters and H a subset of cardinality k . Then, $\text{reg}_{k-\min}(H)$ is unbounded in direction \vec{v} if and only if $\text{reg}_{k-\max}(E \setminus H)$ is unbounded in direction $-\vec{v}$.*

In particular, the Gaussian map of the order- k Voronoi diagram induced by d_{\min} is a reflection at the origin of the Gaussian map of the order- $(n-k)$ Voronoi diagram induced by d_{\max} .

2.7 Conclusion

We derive bounds on the complexity of the order- k Voronoi diagram and its Gaussian map, listed in Table 2.1. The results are tight for the farthest Voronoi diagram. Moreover, we provide an algorithm to compute the Gaussian map of the farthest Voronoi diagram in three dimensions in a worst-case optimal time. It remains an open problem to determine whether or not the lower bounds on the complexity of VD_k and $GM(VD_k)$ for segments, as listed in Table 2.1, extend also to lines, when $k < n - 1$.

There is a gap between our lower and upper bounds on the complexity of the Gaussian map of the order- k Voronoi diagram. What is the correct bound and how can the diagram be constructed efficiently? This question is related to problem 3 in Mitchell and O'Rourke [2001]: “What is the combinatorial complexity of the Voronoi diagram of a set of lines (or line segments) in three dimensions?”

We believe that knowing the structure of the Gaussian map, or equivalently, the structure of d -dimensional cells of the order- k Voronoi diagram, is a fundamental first step that can help in analyzing the whole diagram. It may also be useful in constructing the full diagram. We leave this question for further research.

Chapter 3

On the Trisector of Lines in Three Space

Understanding properties of the edges of the Voronoi diagram is an integral part in describing the complete diagram. In \mathbb{R}^2 the edges are pieces of bisectors, i.e. the set of points in the plane, which are equidistant to two sites.

Definition 3.0.1. For two sites s_1, s_2 , let $B(s_1, s_2)$ denote the bisector of the two sites, i.e. $B(s_1, s_2) = \{ x \in \mathbb{R}^d \mid d(x, s_1) = d(x, s_2) \}$.

For lines and line segments as sites in \mathbb{R}^2 , the bisector of two sites is well studied. It is one connected component, consisting of up to 7 pieces put together. These pieces are rays, line segments and parabolic arcs, see Fig. 3.1. Additionally, the bisector has two unbounded directions. If we consider the distance from a point p on the bisector to one of the sites, then this distance has a global minimum and is monotone increasing as the point p moves along the bisector to infinity.

Observation 3.0.2. The distance function along a bisector to its defining sites has exactly one global minimum and is monotone increasing in both directions.

The bisectors properties, especially the monotonicity, are important ingredients for designing construction algorithms for the Voronoi diagram of the involved sites, as we will see in Section 3.1.

Features of the Voronoi diagram of generalized sites in three and higher dimensions are not that well understood. This is partially due to the increased algebraic difficulties involved.

Definition 3.0.3. For three sites s_1, s_2, s_3 , let $T(s_1, s_2, s_3)$ denote the trisector of the three sites, i.e. $T(s_1, s_2, s_3) = \{ x \in \mathbb{R}^d \mid d(x, s_1) = d(x, s_2) = d(x, s_3) \}$.

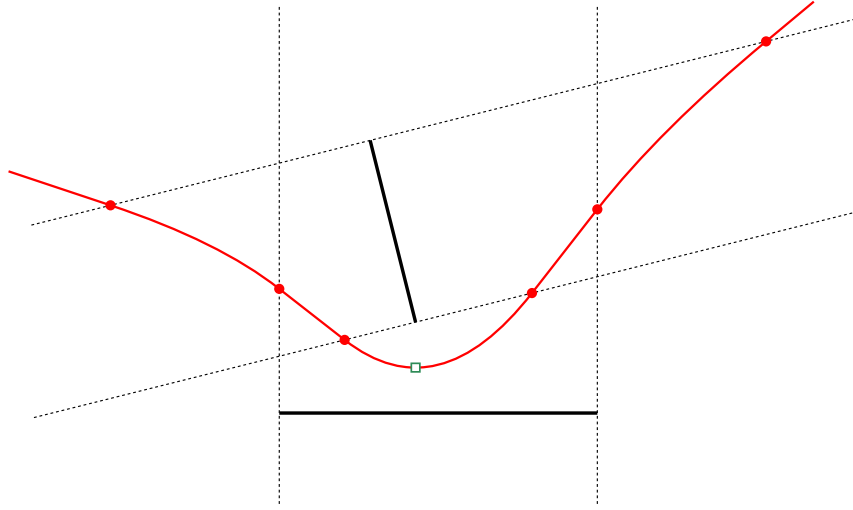


Figure 3.1. The bisector (red) of two line segments (black) in the plane. The red points indicate the ends of the bisector's pieces, i.e. rays, parabolic arcs, and line segments. The minimum distance of the points on the bisector to one of the sites is realized at the green square.

The paper Everett et al. [2009] describe the Voronoi diagram of just three lines in \mathbb{R}^3 , mostly focusing on the trisector of the lines. The trisector consists of up to four unbounded branches, which are solutions of polynomials of degree up to four. These can be a nonsingular quartic or a nonsingular cubic, if the lines are in general position, i.e. they are pairwise skew and not all parallel to a common plane. A trisector of three lines is illustrated in Fig. 3.2.

They also provide an algorithm, which outputs a set of two planes, which isolate one branch of the trisector from the other. We will use this algorithm in order to confirm that three points are on a common branch of the trisector.

One might expect that the branches of the trisector reveal a similar monotonicity property as the bisector in the plane, namely that the distance from a moving point along the trisector to the lines first decreases until it reaches a minimum and then only increases afterwards.

Surprisingly, this is not true and in the following we will show that the distance function along a branch of the trisector can actually have a local maximum.

Theorem 3.0.4. *Consider the trisector of three lines in \mathbb{R}^3 . The distance function along one branch of the trisector can admit a local maximum.*

Proof. The proof idea for this theorem is that we describe three points $P^{(1)}, P^{(2)}, P^{(3)}$ on a trisector. We then show that they all lie on the same branch of the trisector, and in the order $P^{(1)}, P^{(2)}, P^{(3)}$. Finally the distances from the three

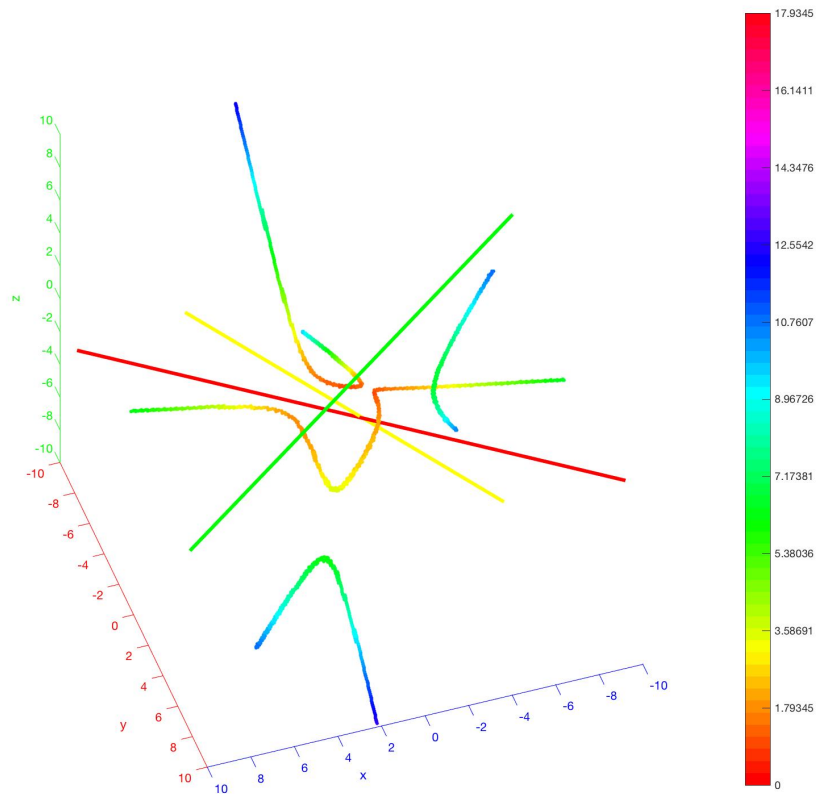


Figure 3.2. The trisector of three lines (red, green, yellow) consists of up to four branches. The colors along the trisector encode the distance to the lines. In this example, the distance function along one branch admits a local maximum, as the color coding changes from orange to yellow and back to orange.

points to the sites s_1, s_2, s_3 satisfy the inequalities $d(P^{(1)}, s_1) < d(P^{(2)}, s_1)$ and $d(P^{(2)}, s_1) > d(P^{(3)}, s_1)$. This implies that there must be a local maximum between the points $P^{(1)}$ and $P^{(3)}$, since the distance function $d(\cdot, s_1)$ is continuous. We choose the three lines in the same fashion as they were chosen in Everett et al. [2009]. With the points

$$\begin{aligned} L_{11} &= (0, 0, 4) & \text{and} & & L_{12} &= (4, 3, 4) \\ L_{21} &= (0, 0, -4) & \text{and} & & L_{22} &= (4, -3, -4) \\ L_{31} &= (1, 1, 0) & \text{and} & & L_{32} &= (4, 1, 4) \end{aligned}$$

we define the three lines as $s_1 := \overline{L_{11}L_{12}}$, $s_2 := \overline{L_{21}L_{22}}$, and $s_3 := \overline{L_{31}L_{32}}$.

The points on the trisector, are all the ones which have equal distance to all three lines, i.e. for $p \in R^3$:

$$d(p, s_1) = d(p, s_2) = d(p, s_3)$$

For the point $p = (p_x, p_y, p_z)$ this corresponds to the following two equations of degree two in its coordinates.

$$\frac{25(p_z - 4)^2 + (3p_x - 4p_y)^2}{25} = \frac{25(p_z + 4)^2 + (4p_y + 3p_x^2)}{25} \quad (3.1)$$

$$\frac{25(p_z - 4)^2 + (3p_x - 4p_y)^2}{25} = \frac{4(p_z - p_x + 1)^2 + (4p_y - 4 - p_z)^2 + (4p_y - 3 - p_x)^2}{33} \quad (3.2)$$

We select the three points $P^{(1)}, P^{(2)}, P^{(3)}$ with the following coordinates:

$$\begin{aligned} P_x^{(1)} &= -4 & P_y^{(1)} &= \frac{55\sqrt{14622}-4425}{524} & P_z^{(1)} &= \frac{33\sqrt{14622}-2655}{655} \\ P_x^{(2)} &= 4 & P_y^{(2)} &= \frac{5\sqrt{5996562}+3575}{1724} & P_z^{(2)} &= -\frac{3\sqrt{5996562}+2145}{2155} \\ P_x^{(3)} &= 8 & P_y^{(3)} &= \frac{10\sqrt{807873}-6825}{596} & P_z^{(3)} &= -\frac{12\sqrt{807873}-8190}{745}. \end{aligned}$$

One can easily verify that the selected points $P^{(1)}, P^{(2)}, P^{(3)}$ indeed lie on the trisector, as their coordinates satisfy the conditions of Eqs. (3.1) and (3.2). Moreover the distances from these points to the lines are

$$\begin{aligned} d(P^{(1)}, s_1) &= \frac{96120142 - 661980\sqrt{14622}}{429025} \\ d(P^{(2)}, s_1) &= \frac{322318742 + 48620\sqrt{5996562}}{4644025} \\ d(P^{(3)}, s_1) &= \frac{332445913 - 333060\sqrt{807873}}{555025} \end{aligned}$$

and indeed satisfy the desired inequalities, i.e. $d(P^{(1)}, s_1) < d(P^{(2)}, s_1)$ and $d(P^{(2)}, s_1) > d(P^{(3)}, s_1)$. We can compute the four critical values with respect to the y -axis in the projection of the trisector to the $x - y$ -plane, which are roughly $c_1 \approx -4.36, c_2 \approx -1.25, c_3 \approx 10.0, c_4 \approx 17.8$. Since the y -coordinates of the three points $P^{(1)}, P^{(2)}, P^{(3)}$ are all between the middle two critical values, the points must all lie on the common trisector branch called C_0 in Everett et al. [2009]. According to Proposition 18 of Everett et al. [2009] and by the choice of the lines, all branches of this trisector are strictly monotone with respect to the x -direction. Since $P_x^{(1)} < P_x^{(2)} < P_x^{(3)}$, the three points also appear in that order along one branch of the trisector. We can conclude that the distance function from a point on the trisector branch C_0 to the sites must therefore admit a local maximum between the points $P^{(1)}$ and $P^{(3)}$. \square

3.1 Review: Farthest Voronoi diagram of lines and line segments in \mathbb{R}^2

We outline a simple $O(n \log n)$ -time algorithm to construct the Euclidean FVD(S) for a set of line segments S in \mathbb{R}^2 discovered by Aurenhammer et al. [2006]. The algorithm is employing a so-called “collapse” strategy: it first computes $\text{GM}(\text{FVD}(S))$, i.e. all unbounded faces, and then gradually constructs the diagram adding edges and vertices in decreasing order in the distance of a feature to its farthest site.

We give a high level description of the algorithm; refer to Fig. 3.3 for an illustration. The algorithm starts at the vertices of $\text{GM}(\text{FVD}(S))$, which are all directions of unbounded edges of $\text{FVD}(S)$. For every pair of edges that are consecutive in circular order, their next intersection point is computed, if one exists. Out of these intersection points, the one with maximum distance to its farthest site is the next vertex of the diagram. This “collapse” event is processed by **(i)** constructing the vertex, **(ii)** constructing the edges leading to this vertex, **(iii)** removing the edges from further consideration, and **(iv)** starting a new edge. At the constructed vertex a face “collapses”, since it is fully constructed and will not be considered again. The new edge is part of the bisector of the two faces neighboring the collapsed face. This procedure, of computing and processing new “collapse” events, is repeated until all the remaining edges intersect in a single point.

The correctness of the collapse algorithm is based on the following three properties:

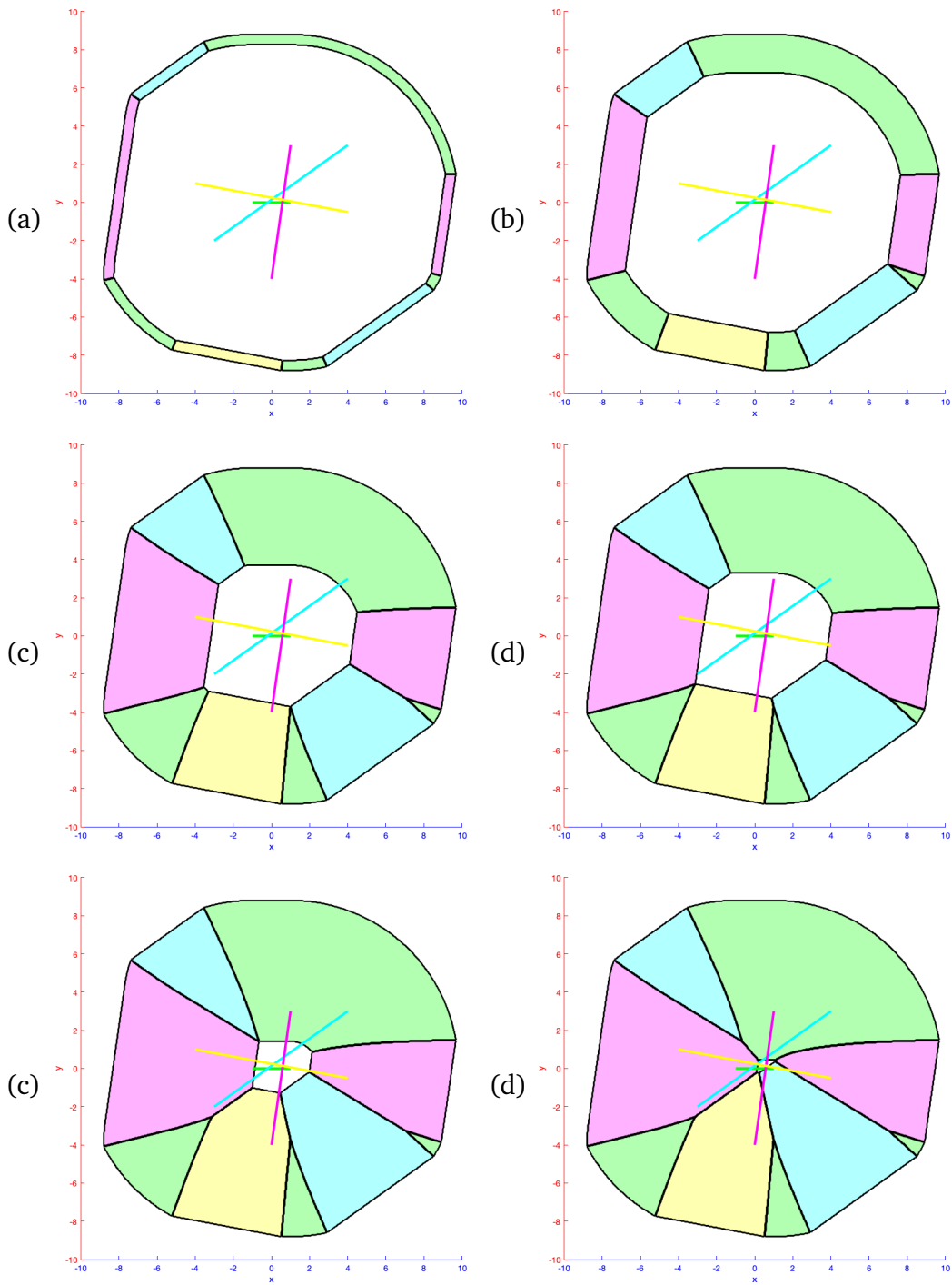


Figure 3.3. Different events during the collapse algorithm for constructing the farthest Voronoi of line segments in \mathbb{R}^2 .

Lemma 3.1.1. *For any positive value $c \in \mathbb{R}$, the set of points with distance c to their farthest line segments in $\text{FVD}(S)$ of a set of segments S forms a cycle.*

Proof. Consider for a line segment $s \in S$ its distance function, which maps any point $p \in \mathbb{R}^2$ to the distance $d(p, s)$. It is well known that this function, which is living in \mathbb{R}^3 , is convex. The farthest Voronoi diagram in turn corresponds to the upper envelope of distance functions of all sites. The upper envelope of convex functions is again convex. The intersection of such a convex function in \mathbb{R}^3 with a horizontal plane $z = c$ is one cycle. By construction, these are all the points of distance c to their farthest site. \square

Lemma 3.1.2. *Every vertex of $\text{FVD}(S)$ of a set of segments S has at least two incident edges with higher distance to the farthest site.*

Proof. Assume there was a vertex v in $\text{FVD}(S)$, for which the distance function to the farthest site is smaller for at least two incident edges e_1, e_2 . Moreover let r_1 (resp. r_2) be the rays, which extend the edges e_1 (resp. e_2) such that there is a smooth transition from the edge to the respective ray. The situation is illustrated in Fig. 3.4. Denote by c the face that is bounded by these two edges and let s be the corresponding farthest site of that face. Let t be the point on s , which realizes the distance between v and s . Due to Lemma 2.2.3, the ray r starting at v and facing away from t is entirely contained in $\text{freg}(s)$. This implies that point t must lie within the wedge bounded by the rays r_1 and r_2 . Therefore the distances from points on e_1 and e_2 to s cannot be both smaller than the distance from v to s , which is a contradiction to our assumption. \square

The next Lemma directly follows from Observation 3.0.2.

Lemma 3.1.3. *In $\text{FVD}(S)$ of a set of segments S , the distance function to the farthest site along an edge has its maximum at a vertex.*

An overview of the collapse algorithm is given in Algorithm 2.

Lemma 3.1.4. *Algorithm 2 constructs $\text{FVD}(S)$ of a set of n segments S in $O(n \log n)$ time.*

Proof. The time complexity of the algorithm is straightforward. The algorithm take $O(n \log n)$ time to compute the $\text{GM}(\text{FVD}(S))$ of n line segments in \mathbb{R}^2 , see Aurenhammer et al. [2006]. Initially there are $O(n)$ unbounded edges and also $O(n)$ many vertex candidates need to be computed. These candidates are sorted according to their distance to the farthest site, which takes another $O(n \log n)$

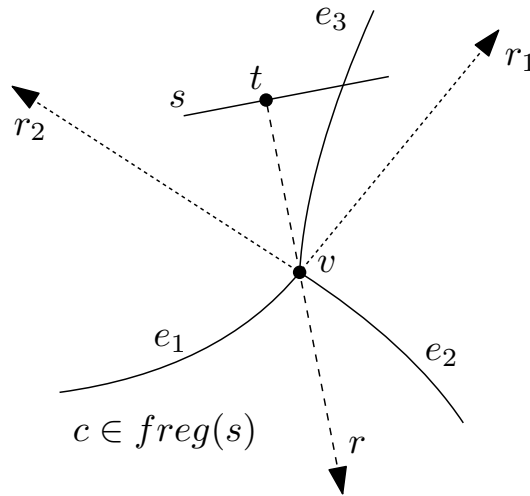


Figure 3.4. There cannot be a vertex v in the farthest Voronoi diagram of lines segments with two incident edges with increasing distance towards v .

time. At each vertex we need to remove two edges of the current list of edges and add a new edge e instead. Additionally the two vertex candidates related to e need to be computed and their distance to their farthest site needs to be compared with the other candidates. This takes $O(\log n)$ time. The FVD(S) has $O(n)$ many vertices, resulting in a total of $O(n \log n)$ time for processing all.

We now prove the correctness of the algorithm. The algorithm constructs the features of the diagram (vertices and edges) in decreasing distance to their farthest site. First note, that because of Lemma 3.1.1, the set of points with same distance to their farthest site form a single cycle. So to keep track of the edges, which are “*under construction*”, it suffices to use a circular list.

Next we argue that the algorithm correctly finds the next vertex, assuming that all vertices with bigger distance have already been constructed. By Lemma 3.1.2, every vertex has at least two edges with decreasing distance towards it. Thus, the next vertex is the intersection of a pair of edges, which are under construction. Further the two involved edges need to be consecutive, as otherwise, the cyclic property of Lemma 3.1.1 would be violated. Since we are constructing the diagram in decreasing distance, the algorithm picks the candidate vertex with biggest distance.

At each vertex event, the algorithm starts constructing a new edge. The algorithm does not miss any new edges because edges can only start at vertices. This is due to the distance function on an edge being monotone and not exhibiting local maximum; see Lemma 3.1.3. \square

Algorithm 2: Collapse algorithm

Input : Set of n line segments S
Output : FVD(S)

- 1 Compute GM(FVD(S));
- 2 Save the unbounded edges of FVD(S) in a cyclic list L ;
- 3 **while** $|L| > 2$ **do**
- 4 **for** every pair of consecutive edges e_1, e_2 in L **do**
- 5 | Compute the intersection of e_1 and e_2 ;
- 6 $v \leftarrow$ intersection with biggest distance to the farthest site;
- 7 Add v and its incident edges to the diagram;
- 8 Update the edges in L ;

3.2 Towards constructing the FVD of lines and line segments in \mathbb{R}^3

The collapse algorithm seems to be generalizing to three dimensional space, i.e. first computing GM(FVD(S)) for a set of lines and line segments S , and then gradually constructing the diagram adding faces, edges and vertices in decreasing order in the distance of a feature to its farthest site. In the two dimensional version, the construction takes $O(\log n)$ time per vertex of the diagram. But one key property of the collapse idea in \mathbb{R}^2 is that new edges are quickly discovered, where already constructed edges intersect. Unfortunately, this property is not satisfied in \mathbb{R}^3 . The edges of FVD(S) are pieces of trisectors and due to Theorem 3.0.4, the edges of FVD(S) might need to be discovered at their interior of the edge. In some sense, new edges might appear without any warning.

A possible solution for this issue would be to pre-compute all possible trisectors and find all local maxima of the distance function in the sense of Theorem 3.0.4. This would introduce an additional $\Theta(n^3)$ time during the initialization. Moreover the algorithm would need to figure out, which of these local maxima actually appear in the diagram and how to localize these within the constructed pieces of the diagram.

Chapter 4

Farthest Polyhedral Voronoi diagram

This chapter is based on the following paper:

- Franz Aurenhammer, Evanthia Papadopoulou, and Martin Suderland. [2021] Piecewise-linear farthest-site Voronoi diagrams. *32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

For most generalized Voronoi diagrams, the partition of space they define is not piecewise linear any more, but is rather composed of curved geometric objects of various dimensions and shapes. This complicates their structural analysis as well as their computational construction, especially in dimensions higher than two (where results are becoming comparatively sparse). For instance, already in three-dimensional space \mathbb{R}^3 , the algebraic description of the edges and facets of the Euclidean Voronoi diagram of straight lines becomes exceedingly complicated Everett et al. [2009]. What is more, the combinatorial complexity of this diagram is a major open problem in computational geometry Mitchell and O'Rourke [2001]. There is a gap of an order of magnitude between the $\Omega(n^2)$ lower bound Aronov [2002] and the only known upper bound of $O(n^{3+\varepsilon})$, for any $\varepsilon > 0$ Sharir [1994].

Certain types of Voronoi diagrams retain their piecewise linear structure, however. For example, the so-called power diagram Aurenhammer [1987] has this property. Another prominent class, and the one of interest in the present note, is induced by (convex) *polyhedral distance functions*. Intuitively speaking, the distance from a point x to a site s is now measured as the extent at which a given convex polyhedron \mathcal{P} , which being centered at x , has to expand till it starts touching s .

Several authors succeeded in deriving strong bounds on the combinatorial complexity of such Voronoi diagrams. If the n sites are points and the distance polytope \mathcal{P} is a simplex or a cube—the latter just giving the L_∞ distance—then this complexity in \mathbb{R}^d is $\Theta(n^{\lceil \frac{d}{2} \rceil})$, see Boissonnat et al. [1998]. (The dimension d is considered a constant throughout this paper.) In \mathbb{R}^3 , the same bound still applies when any constant-sized convex polytope is chosen for \mathcal{P} Icking and Ma [2001]. For the sites being n straight lines in \mathbb{R}^3 , with \mathcal{P} defined as before, near-quadratic bounds of $\Omega(n^2\alpha(n))$ and $O(n^2\alpha(n)\log(n))$ can be obtained Chew et al. [1998]. Here $\alpha(n)$ is the extremely slowly growing inverse Ackermann function. If we consider as sites disjoint convex polyhedra with n faces in total, then the complexity is $O(n^{2+\epsilon})$, as has been shown in Koltun and Sharir [2004]; this sharpens to $O(n^2\alpha(n)\log(n))$ if all the sites are line segments.

Though Voronoi diagrams for convex polyhedral distance functions—in comparison to Euclidean Voronoi diagrams—thus proved easier to deal with concerning their combinatorial aspects, this does not seem to carry over to their algorithmic aspects. In fact, the papers cited above do not provide algorithms for computing such diagrams, and we are not aware of any construction algorithm particular to them.

As we shall show, the situation changes if the so-called *farthest-site* variant of the diagram is considered (rather than the *closest-site* variant as above). We will show that the complexity of this diagram in \mathbb{R}^d is $\Theta(n^{\lceil \frac{d}{2} \rceil})$ in the worst-case, and that it can be computed in optimal time $O(n^{\lceil \frac{d}{2} \rceil} + n \log n)$, mainly by using higher-dimensional convex hull algorithms. This result holds under rather general conditions: Sites can be arbitrary convex polyhedra (which may be unbounded or overlapping, having a total of n faces of various dimensions), the distance polytope \mathcal{P} may be unbounded (though constant-sized), and the resulting distance function can be additively and/or multiplicatively weighted for each site. Moreover, each site may get allotted a particular distance polytope, in order to generate an anisotropic scenario where sites can influence their surrounding in an individual way.

Unfortunately, Euclidean farthest-site Voronoi diagrams have their peculiarities (unless all sites are points, in which case their combinatorial and computational behavior is much like their closest-point counterparts Seidel [1987]). Their regions may disconnect into a large number of nonconvex parts, and the close relationship between nonempty regions and the convex hull of the sites is lost; see Aurenhammer et al. [2006] for line segment sites in \mathbb{R}^2 , and Papadopoulou and Dey [2013] for a generalization to arbitrary L_p -metrics. The only result for non-point sites in higher dimensions we are aware of is Barequet et al. [2019],

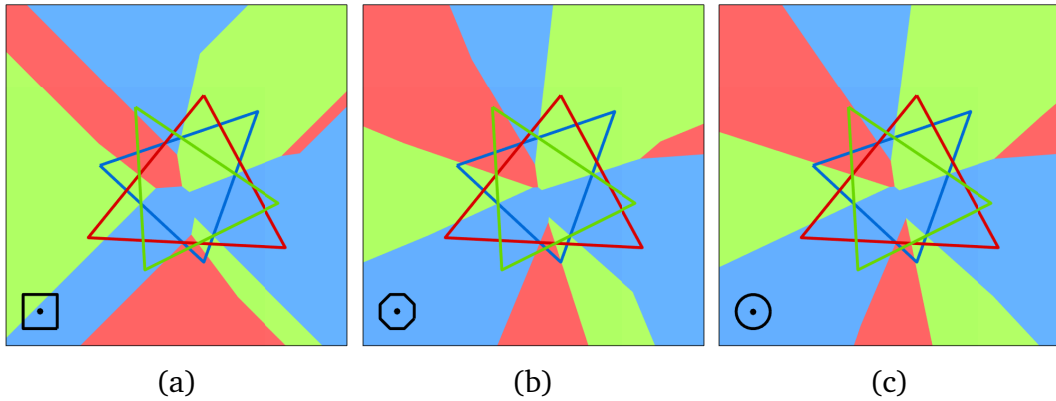


Figure 4.1. Two approximations (a) and (b) of a Euclidean farthest-site Voronoi diagram (c). The sites are three overlapping triangles. Their boundaries are visualized in individual colors, and their farthest regions are painted accordingly. The distance polygons used—a square in (a) and a regular 8-gon in (b)—are shown in the bottom-left corner.

who derive structural and combinatorial properties for the farthest-site diagram of lines and line segments in \mathbb{R}^d . They characterize its unbounded cells, of which there are up to $\Theta(n^{d-1})$ many in the worst-case, and describe an algorithm to compute these in near-optimal time.

With our results, a large class of Euclidean farthest-site Voronoi diagrams for convex sites in \mathbb{R}^d , even in their weighted and/or anisotropic variants, can be approximated in a piecewise-linear manner, and are computable by a simple and uniform approach: In \mathbb{R}^3 for example, being probably the most interesting case, the Euclidean ball can be β -approximated by a convex polytope \mathcal{P} with $O(1/\beta)$ vertices Koltun and Sharir [2004], such that the convex distance induced by \mathcal{P} is at most $1 + \beta$ times the Euclidean distance. As a particularly useful result, a simple method for computing a piecewise-linear approximation of size $O(n^2)$ of the Euclidean farthest-site Voronoi diagram for lines and/or line segments in \mathbb{R}^3 becomes available. Even the planar instance is interesting: The fastest known algorithm for the Euclidean farthest-site Voronoi diagram for polygonal sites in \mathbb{R}^2 runs in time $O(n \log^3 n)$ Cheong et al. [2011], whereas our approximation can be computed in time $O(n \log n)$ if the polygonal sites are convex. Figure 4.1 illustrates the similarity between these diagrams.

4.1 Convex polyhedral distance

We define a *polyhedron* in \mathbb{R}^d as the nonempty and finite (but possibly unbounded) intersection of closed halfspaces of \mathbb{R}^d . Note that a polyhedron does not need to be full-dimensional: For example lines, line segments, and single points are included as lower-dimensional instances.

Any d -dimensional polyhedron \mathcal{P} which contains the origin in its interior can be used to define a so-called *convex polyhedral distance*, from a point $x \in \mathbb{R}^d$ to a point $q \in \mathbb{R}^d$:

$$\delta_{\mathcal{P}}(x, q) = \inf_{t \geq 0} \{ t \mid q \in x + t \cdot \mathcal{P} \}.$$

In other words, $\delta_{\mathcal{P}}(x, q)$ describes the amount $t \geq 0$ by which \mathcal{P} , when being placed at x , has to be scaled so as to cover q ; see Fig. 4.2. Note that $\delta_{\mathcal{P}}$ is a directed distance. We shall call \mathcal{P} the *distance polytope* that induces the distance function $\delta_{\mathcal{P}}$.

Let $\mathcal{P}^R = \{ -p \mid p \in \mathcal{P} \}$ denote the reflection of the distance polytope about the origin.

Observation 4.1.1. *We have $\delta_{\mathcal{P}}(x, q) = \delta_{\mathcal{P}^R}(q, x)$.*

Proof. Suppose that $q \in x + t \cdot \mathcal{P}$. Then there is a point $p \in \mathcal{P}$ with $q = x + t \cdot p$, that is, $x = q - t \cdot p$. Thus we have $x \in q - t \cdot \mathcal{P}$, which by the identity $-t \cdot \mathcal{P} = t \cdot \mathcal{P}^R$ means $x \in q + t \cdot \mathcal{P}^R$. \square

Consider a set S of point sites in \mathbb{R}^d , and identify \mathbb{R}^d with the hyperplane $x_{d+1} = 0$ in $(d+1)$ -dimensional space \mathbb{R}^{d+1} . Observation 4.1.1 suggests to associate the distance polytope \mathcal{P} with a *distance cone* $C_{\mathcal{P}}$ in \mathbb{R}^{d+1} , such that $C_{\mathcal{P}}$ reflects with its height the polyhedral distance induced by \mathcal{P} .

$$C_{\mathcal{P}} = \bigcup_{t \geq 0} \begin{pmatrix} t \cdot \mathcal{P}^R \\ t \end{pmatrix} \quad (4.1)$$

$C_{\mathcal{P}}$ is a polyhedral cone obtained from scaling the reflected polytope \mathcal{P}^R . Its apex is at the origin. Let $C_{\mathcal{P}}(q_i)$ be the translate of $C_{\mathcal{P}}$ with its apex at some point site $q_i \in S$. Then for any point $x \in \mathbb{R}^d$, the $(d+1)^{\text{st}}$ coordinate (called *height*) of the vertical projection of x to $C_{\mathcal{P}}(q_i)$ equals the distance $\delta_{\mathcal{P}}(x, q_i)$.

Let now, more generally, the set S consist of polyhedral sites s_i in \mathbb{R}^d . We construct for each site $s_i \in S$ a distance cone as follows. Take the Minkowski sum $s_i \oplus C_{\mathcal{P}}$. (The *Minkowski sum* of point sets A and B is defined as $A \oplus B =$

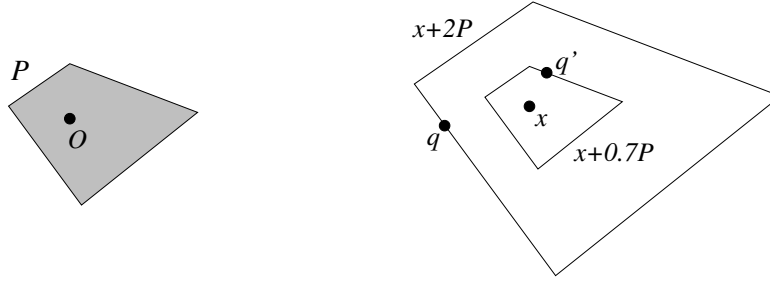


Figure 4.2. Polyhedral distance induced by \mathcal{P} : $d_{\mathcal{P}}(x, q) = 2$ and $d_{\mathcal{P}}(x, q') = 0.7$

$\{a + b \mid a \in A \wedge b \in B\}$.) Because the Minkowski sum of two convex polyhedra is again a convex polyhedron, the object $s_i \oplus C_{\mathcal{P}}$ is the intersection of halfspaces of \mathbb{R}^{d+1} . One of them is bounded from below by the hyperplane $x_{d+1} = 0$ (if the site s_i is full-dimensional). We ignore this halfspace, and intersecting the remaining ones we obtain an unbounded polyhedron in \mathbb{R}^{d+1} , which we call the *distance cone* of s_i , and denote with $C_{\mathcal{P}}(s_i)$.

$C_{\mathcal{P}}(s_i)$ exhibits the following useful properties.

- For the special case of s_i being a point site q_i , the definition of $C_{\mathcal{P}}(s_i)$ is consistent with that of $C_{\mathcal{P}}(q_i)$ before.
- Let $d_{\mathcal{P}}(x, s_i)$ be the height of the vertical projection of a point $x \in \mathbb{R}^d$ to $C_{\mathcal{P}}(s_i)$. If x does not lie in the interior of s_i , then $d_{\mathcal{P}}(x, s_i)$ is non-negative and equals the polyhedral distance of x to s_i , which is commonly defined as

$$\delta_{\mathcal{P}}(x, s_i) = \inf_{t \geq 0} \{ t \mid x + (t \cdot \mathcal{P}) \cap s_i \neq \emptyset \}.$$

- If x lies in the interior of s_i then $d_{\mathcal{P}}(x, s_i)$ is negative, and measures how much x is inside the (full-dimensional) polyhedral site s_i by taking the minimum polyhedral distance to its facets; see Fig. 4.1. This is because the part of $C_{\mathcal{P}}(s_i)$ that lies below the hyperplane $x_{d+1} = 0$ is determined solely by halfspaces which stem from s_i (and not from $C_{\mathcal{P}}$ in Formula (4.1)). That is, $d_{\mathcal{P}}$ is related to a generalized medial axis of s_i in this case.

4.2 Farthest-site Voronoi diagram

The so-called *farthest-site Voronoi diagram* of a set S of sites in \mathbb{R}^d , for short FVD(S), is a partition of \mathbb{R}^d into regions such that all points within a fixed region

have the same farthest site. As before, let the sites s_i in S be polyhedra. These may be of any dimension k , for $0 \leq k \leq d$, and are not required to be disjoint or bounded.

We are interested in the diagram $\text{FVD}(S)$ induced by the convex polyhedral distance function $d_{\mathcal{P}}$ in Section 4.1, for a given distance polytope \mathcal{P} . Being a farthest-site diagram, $\text{FVD}(S)$ corresponds to the pointwise maximum of the functions $d_{\mathcal{P}}(x, s_i)$, for $s_i \in S$, on \mathbb{R}^d . $\text{FVD}(S)$ thus corresponds to the *upper envelope* of the boundaries of the distance cones $C_{\mathcal{P}}(s_i)$ that define these functions, which, in turn, is given by the common intersection of these cones. Let us formulate this result in the following way.

Theorem 4.2.1. *Let I be the (unbounded) convex polyhedron in \mathbb{R}^{d+1} that results from intersecting the distance cones $C_{\mathcal{P}}(s_i)$, for all sites $s_i \in S$. Then $\text{FVD}(S)$ is the vertical projection of I onto the hyperplane $x_{d+1} = 0$ of \mathbb{R}^{d+1} .*

One of the consequences of Theorem 4.2.1 is that $\text{FVD}(S)$ is a piecewise linear diagram. Each region of $\text{FVD}(S)$ is pre-partitioned into convex polyhedra (the projected facets of I), and these regions define a partition of \mathbb{R}^d . Let us point out that, in earlier papers on Voronoi diagrams for polyhedral distance functions (e.g. in Koltun and Sharir [2004]), the distance of a point x to a site was set to zero in case x falls in the interior of that site. As a consequence, when the sites are not chosen to be pairwise disjoint, the part of \mathbb{R}^d covered by their union does not get partitioned by the diagram. Our more general definition of polyhedral distance, via distance cones, remedies this shortcoming.

The combinatorial complexity of $\text{FVD}(S)$ is given by that of the projection polyhedron I in \mathbb{R}^{d+1} . I is the intersection of distance cones, and each distance cone $C_{\mathcal{P}}(s_i)$, in turn, is the intersection of halfspaces of \mathbb{R}^{d+1} . It is clear from Section 4.1 that the number of such halfspaces per cone is bounded by the number of facets of the Minkowski sum $s_i \oplus \mathcal{P}^R$, for the reflected distance polytope \mathcal{P}^R . A single face of s_i , combined with a single face of \mathcal{P}^R , can yield at most one facet of $s_i \oplus \mathcal{P}^R$; see e.g. Das and Sarvottamananda [2018]. Therefore, if we assume that \mathcal{P} (and with it \mathcal{P}^R) is of constant size, and that s_i has a total of n_i faces of different dimensions, then $C_{\mathcal{P}}(s_i)$ is defined by $O(n_i)$ halfspaces of \mathbb{R}^{d+1} . In conclusion, when putting $n = \sum_{s_i \in S} n_i$, the polyhedron I is the intersection of $O(n)$ halfspaces of \mathbb{R}^{d+1} , and its complexity is bounded from above by $O(n^{\lceil \frac{d}{2} \rceil})$ (provided $d = O(1)$), by the well-known upper bound theorem. We will show in Section 4.3 that this complexity can be asymptotically attained in the worst case. Observe that I has $O(n)$ facets, and that $\text{FVD}(S)$ thus has this very number of full-dimensional cells.

Concerning computational aspects, the halfspaces defining a particular cone $C_{\mathcal{P}}(s_i)$ can be singled out by (basically) computing the Minkowski sum $s_i \oplus \mathcal{P}^R$. This can be done Das and Sarvottamananda [2018], for instance, by pairwise adding up the $O(n_i)$ vertices of s_i and the $O(1)$ vertices of \mathcal{P}^R , and computing the convex hull of the resulting $O(n_i)$ points in \mathbb{R}^d , spending a total of $O(n^{\lfloor \frac{d}{2} \rfloor} + n \log n)$ time for all sites $s_i \in S$, when the optimal convex hull algorithm in Chazelle [1993] is applied. The construction of the projection polyhedron I is more time-consuming and takes $O(n^{\lfloor \frac{d}{2} \rfloor} + n \log n)$ time; we use the convex hull algorithm in Chazelle [1993] again, but now for intersecting $O(n)$ halfspaces in \mathbb{R}^{d+1} .

We may summarize as follows:

Theorem 4.2.2. *Let S be a set of arbitrary polyhedral sites in \mathbb{R}^d , with a total combinatorial complexity of n . The farthest-site Voronoi diagram $\text{FVD}(S)$ of S under the convex distance function induced by a polytope of constant size is of complexity $O(n^{\lfloor \frac{d}{2} \rfloor})$, and it can be computed in $O(n^{\lfloor \frac{d}{2} \rfloor} + n \log n)$ time. The number of d -dimensional cells of $\text{FVD}(S)$ is bounded by $O(n)$.*

The dependence on d of the bounds stated above is the same as for convex hulls of finite point sets.

4.3 More properties of FVD

The maximal size of farthest-site diagrams may be much smaller than that of their closest-site counterparts; several examples can be found in Aurenhammer et al. [2013]. The question arises whether the upper bound given in Theorem 4.2.2 is asymptotically tight.

For special sets of polyhedral sites in \mathbb{R}^d , the diagram $\text{FVD}(S)$ is indeed small, namely, when the sites in S have only a constant number of orientations. Then the halfspaces defining the projection polyhedron I in \mathbb{R}^{d+1} will have only a constant number of orientations as well, and all but $O(1)$ of them will be redundant because their bounding hyperplanes are parallel. Consequently, the polyhedron I and its projection $\text{FVD}(S)$ will be of constant size, and can be found in $O(n)$ time.

Observe that the case of S being a set of n point sites in \mathbb{R}^d is covered above, because each point site can be described by the intersection of $d+1$ halfspaces of \mathbb{R}^d , having the same fixed orientations. Not included, however, is the case of n line segment sites in \mathbb{R}^d , because the $d+2$ halfspaces describing a line segment will be of different orientation for different sites, in general. In fact, sites of very simple shape can induce large diagrams, as is shown below.

Lemma 4.3.1. *There exists a set S of n sites in \mathbb{R}^d of constant description such that the diagram $\text{FVD}(S)$ has a complexity of $\Omega(n^{\lceil \frac{d}{2} \rceil})$.*

Proof. There exist two hyperplanes h_1, h_2 in \mathbb{R}^{d+1} , and two point sets $Y_1 \subset h_1$ and $Y_2 \subset h_2$ each of size $\frac{n}{2}$, such that the lower convex hull of $Y_1 \cup Y_2$ has $\Theta(n^{\lceil \frac{d+1}{2} \rceil}) = \Theta(n^{\lceil \frac{d}{2} \rceil})$ complexity; this follows from Corollary 12 in Karavelas et al. [2013]. W.l.o.g., we may assume (by applying an affine transformation) that

$$\begin{aligned} h_1 : \sum_{i=1}^d x_i = 1, \quad h_2 : \sum_{i=1}^{d-1} x_i - x_d = 1, \quad \text{and} \\ Y_1 \subset \mathbb{R} \times \left(0, \frac{1}{d}\right]^d, \quad Y_2 \subset \mathbb{R} \times \left(0, \frac{1}{d}\right]^{d-1} \times \left[-\frac{1}{d}, 0\right), \end{aligned} \quad (4.2)$$

which fixes the hyperplanes and guarantees that most coordinates in $Y_1 \cup Y_2$ are small. We now choose a distance polytope \mathcal{P} and a set $S = S_1 \cup S_2$ of sites such that the projection polyhedron I for $\text{FVD}(S)$ is dual to the lower convex hull of $Y_1 \cup Y_2$.

Let the hypercube $[-1, 1]^d$ serve as \mathcal{P} . The set $S_1 \cup S_2$ will consist of n halfspace sites in \mathbb{R}^d . For S_1 , each of its halfspaces s is constructed from a point $y = (y_1, \dots, y_{d+1}) \in Y_1$. In particular, we describe s by the inequality $\sum_{i=1}^d a_i x_i \leq b$, where

$$a_1 = 1, \quad a_i = \frac{y_i}{1 - \sum_{j=2}^d y_j} \quad \text{for } i = 2, \dots, d, \quad \text{and } b = \frac{y_{d+1}}{1 - \sum_{j=2}^d y_j}. \quad (4.3)$$

Note that all a_i and b are positive because of our assumption (4.2). Moreover, we have $\mathcal{P}^R = \mathcal{P}$. Therefore, the Minkowski sum $s \oplus \mathcal{P}^R$ is just a translate of s by the vector $(1, 1, \dots, 1)^T$ in \mathbb{R}^d , which implies that the distance cone $C_{\mathcal{P}}(s)$ is a single halfspace in \mathbb{R}^{d+1} , bounded from below by the hyperplane

$$x_{d+1} = \frac{1}{A} \cdot \left(\sum_{i=1}^d a_i x_i - b \right), \quad \text{where } A = \sum_{j=1}^d a_j. \quad (4.4)$$

By a well-known duality transform, the hyperplane in (4.4) is dual to the point

$$q = (q_1, \dots, q_{d+1}) = \frac{1}{A} \cdot (a_1, \dots, a_d, b)$$

in \mathbb{R}^{d+1} . Substituting the values in (4.3) and simple calculations give

$$q_i = y_i \quad \text{for } i = 2, \dots, d+1, \quad \text{and } q_1 = 1 - \sum_{j=2}^d y_j. \quad (4.5)$$

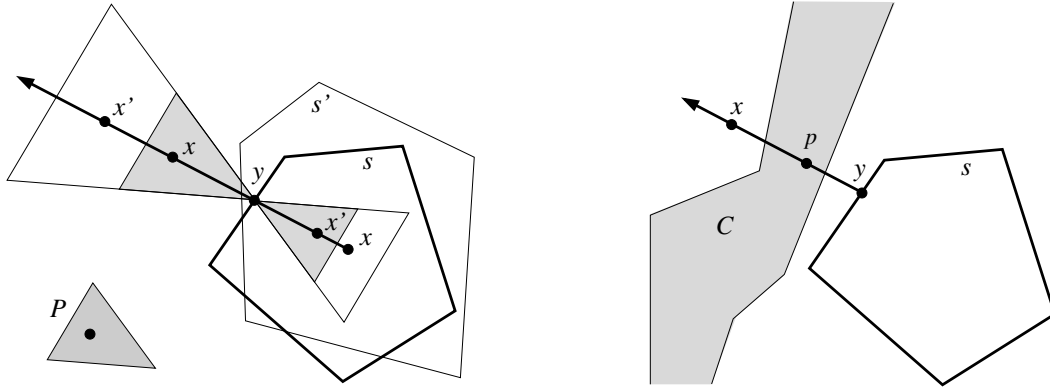


Figure 4.3. Illustrations of the proofs of Lemma 4.3.2 (left) and Lemma 4.3.3 (right).

But this implies $q = y$, because both points lie in the hyperplane h_1 in (4.2): We have $y \in h_1$ by assumption, and $q \in h_1$ by (4.5).

In a similar manner, we can construct suitable halfspace sites s for S_2 from the points y in Y_2 . (We omit these details here.) In conclusion, the projection polyhedron I , being the intersection of all the sites' distance cones, is the intersection of the upper halfspaces $C_{\mathcal{P}}(s)$ for all $s \in S_1 \cup S_2$, and thus I is dual to the lower convex hull of $Y_1 \cup Y_2$. \square

By Lemma 4.3.1, the runtime in Theorem 4.2.2 is asymptotically optimal in the worst case for $d \geq 3$. For $d = 2$, a reduction from sorting proves optimality.

Being the projection of I , the diagram $\text{FVD}(S)$ is a polyhedral cell complex in \mathbb{R}^d which is face-to-face. Its *cells* (polyhedra of dimension d) are nonconvex in general, as are its *facets* (polyhedra of dimension $d-1$). Since the distance polytope \mathcal{P} is (more or less) an approximation of the Euclidean ball, quite a few properties of the Euclidean farthest-site diagram of S carry over to $\text{FVD}(S)$; see e.g. Aurenhammer et al. [2006]; Barequet et al. [2019]. For example, the *region* of a site $s_i \in S$ in $\text{FVD}(S)$ (the set of all points in \mathbb{R}^d being farthest from s_i) is disconnected in general, and it may consist of various cells of $\text{FVD}(S)$. Moreover, the following properties of the cells are preserved.

Lemma 4.3.2. *All cells of $\text{FVD}(S)$ are unbounded, and cells cannot contain voids of any dimension.*

Proof. Let C be some cell of $\text{FVD}(S)$, and assume that C is part of the region of the site $s \in S$. The assertion of the lemma can be easily derived from the following fact: Let x be an arbitrary point in C , and consider the point y on the boundary

of s that realizes the polyhedral distance $d_{\mathcal{P}}(x, s)$. Then the infinite ray r that starts at x and is directed away from y is totally contained in C .

To prove this fact, refer to Figure 4.3 (left). Assume first that $x \notin s$. Then $t = d_{\mathcal{P}}(x, s) \geq 0$, and the homothet $H = x + t \cdot \mathcal{P}$ touches s at y . Since s is the site in S farthest from x , H intersects all the other sites. Let now x' be any point on r such that x lies between x' and y . Put $t' = d_{\mathcal{P}}(x', s)$. Then $H' = x' + t' \cdot \mathcal{P}$ touches s at y too, and H is covered by H' , which implies that H' intersects all other sites as well. This implies that x' lies in the region of s .

If $x \in s$, on the other hand, then $t = d_{\mathcal{P}}(x, s) < 0$, and we have $t \cdot \mathcal{P} = u \cdot \mathcal{P}^R$ with $u = -t > 0$, for the reflected polytope \mathcal{P}^R . The homothet $H = x + u \cdot \mathcal{P}^R$ touches s at y , and since s is farthest from x , H is contained in all other sites now. For any point x' on r between x and y , and $u' = -d_{\mathcal{P}}(x', s)$, $H' = x' + u' \cdot \mathcal{P}^R$ touches s at y again, but is contained in H now and therefore also in all other sites. So x' has to lie in the region of s .

In summary, we conclude that the entire ray r lies in the cell C of the region of s . \square

Let us define the $(d-1)$ -skeleton of $\text{FVD}(S)$ as the union of all the facets of $\text{FVD}(S)$. This skeleton can be disconnected, as a simple construction with only two sites shows; see Figure 4.4(a): Let site s_1 be some polyhedron which approximates a line segment, and take as site s_2 any polyhedron which contains the segment's midpoint but none of its endpoints. Then the region of s_1 disconnects the $(d-1)$ -skeleton of $\text{FVD}(\{s_1, s_2\})$. On the other hand, by the same argument as in Barequet et al. [2019], the following holds:

Lemma 4.3.3. *The $(d-1)$ -skeleton of $\text{FVD}(S)$ is connected, provided that the sites in S are pairwise disjoint.*

Proof. Assume that this skeleton is not connected; see Figure 4.3 (right). Then there exists some cell C of $\text{FVD}(S)$ that splits the skeleton into at least two parts. Let s be the farthest site corresponding to C . The site s does not touch the boundary of C , because of our assumption on the disjointness of the sites. Thus there exists some point $x \notin C$ which is separated from s by C . Let y be the point on s that realizes the polyhedral distance from x to s . By construction, the line segment \overline{xy} intersects C , and we choose a point p in this intersection. Now, by the reasoning in the proof of Lemma 4.3.2, the infinite ray emanating from p in direction x is entirely contained in C . But this implies $x \in C$, which is a contradiction. \square

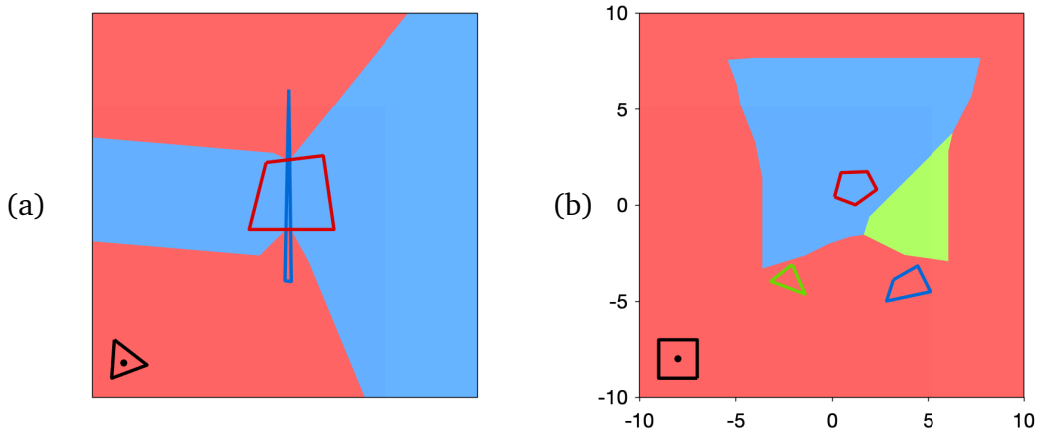


Figure 4.4. (a) The $(d-1)$ -skeleton can be disconnected for non-disjoint sites. (b) A weighted farthest Voronoi diagram of three sites: The blue quadrangle has an additive weight of -1 , and the red pentagon has a multiplicative weight of $\frac{1}{2}$.

4.4 Variants

In certain applications, a model of Voronoi diagram is required where the sites are capable of influencing their surrounding in an individual way; see Aurenhammer et al. [2013]; Okabe et al. [2009] for comprehensive treatments of this topic. One way to achieve this goal is to assign so-called *weights* to the sites, which affect the underlying distance function in an additive and/or multiplicative way.

Let each site $s_i \in S$ have assigned two real numbers $a(s_i)$ and $m(s_i) > 0$, and consider the weighted polyhedral distance:

$$\frac{d_{\mathcal{P}}(x, s_i)}{m(s_i)} - a(s_i)$$

In contrast to the nearest version, the sites' regions in the farthest Voronoi diagram shrink with increasing weights. Interestingly, and unlike the situation for the Euclidean farthest-site diagram, the FVD(S) induced by this distance is still a piecewise-linear cell complex. This becomes evident when the respective distance cones are considered: Additive weighting results in a vertical shift of these cones by an amount of $a(s_i)$, and multiplicative weighting enlarges by a factor of $m(s_i)$ the value $\tan \alpha_j$ of the dihedral angles α_j of aperture of a cone's facets. In particular, each distance cone still is the intersection of $O(n_i)$ halfspaces of \mathbb{R}^{d+1} when site s_i is of complexity n_i .

Multiplicative weighting leads to the occurrence of bounded regions in $\text{FVD}(S)$, as simple examples show (Figure 4.4 (b)). However, purely additive weighting preserves the properties listed in Lemma 4.3.2. In particular, all cells are still unbounded: All facets of the projection polyhedron I for the unweighted $\text{FVD}(S)$ are unbounded, and this fact cannot be altered by vertically shifting any of its defining halfspaces.

We may push things even further, and create an *anisotropic* scenario by allotting an individual distance polytope \mathcal{P}_i to each site s_i . In this way, each site is able to ‘interpret’ its surrounding space in its own way—a concept useful in many situations Aurenhammer et al. [2013]. In fact, the multiplicative weighting scheme is just the special case where $\mathcal{P}_i = m(s_i) \cdot \mathcal{P}$.

In all the extensions above, the properties of the distance cones needed for Theorem 4.2.2 to hold are preserved. We obtain the following general result:

Theorem 4.4.1. *Theorem 4.2.2 remains valid for all the weighted and anisotropic variants of $\text{FVD}(S)$ described above.*

Note finally that all these extensions can be combined, and lead to a very general class of easy-to-compute piecewise-linear farthest-site Voronoi diagrams in \mathbb{R}^d , where the impact of each site can be tuned by its shape, its weights, and its distance polytope including the choice of the polytope’s center.

4.5 Conclusion

Even though the Euclidean distance measure is more intuitive, the polyhedral version has many advantages in the context of Voronoi diagrams of polyhedral sites. Foremost, the features of the diagram are all piecewise linear, which reduces problems regarding the algebraic complexity. Recall that already the trisector of three lines is non-trivial. Moreover, the Euclidean distance can still be well approximated by the distance induced by an appropriate polyhedron.

Finally, for the farthest polyhedral Voronoi diagram we were able to reduce the diagram to the convex hull of points, hence discovering a worst-case optimal-time algorithm for its construction. This result is extremely general as it allows a wide variety of polyhedral sites and distance measure (possibly individual ones for each site), arbitrary additive and multiplicative weights and also works in any dimensions.

Chapter 5

Fermat point and n -ellipses

This chapter is based on the following paper:

- Kolja Junginger, Ioannis Mantas, Evanthia Papadopoulou, Martin Suderland, and Chee Yap. [2021]
Certified approximation algorithms for the Fermat point and n -ellipses.
29th Annual European Symposium on Algorithms, ESA 2021, Vol. 204 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 54:1–54:19.

5.1 Introduction

A classic problem in Facility Location, see e.g., Fekete et al. [2005]; Ostresh Jr [1978], is the placement of a facility to serve a given set of demand points or customers so that the total transportation costs are minimized. The total cost at any point is interpreted as the sum of the distances to the demand points. The point that minimizes this sum is called the *Fermat Point*; see Fig. 5.1. This is an old geometric problem that has inspired scientists over the last three centuries.

A *weighted foci set* is a non-empty finite set of (demand) points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ in \mathbb{R}^d associated with a positive weight function $w : A \rightarrow \mathbb{R}_{>0}$. Each $\mathbf{a} \in A$ is called a *focus* with weight $w(\mathbf{a})$. Let $W := \sum_{\mathbf{a} \in A} w(\mathbf{a})$. The *Fermat distance function* of A is given by

$$\varphi(\mathbf{x}) := \sum_{\mathbf{a} \in A} w(\mathbf{a}) \|\mathbf{x} - \mathbf{a}\|,$$

where $\|\cdot\|$ is the Euclidean norm in \mathbb{R}^d . The global minimum value of φ is called the *Fermat radius* of A , denoted by $r^* = r^*(A)$. Any point $\mathbf{x} \in \mathbb{R}^d$ that achieves this minimum, $\varphi(\mathbf{x}) = r^*$, is called a *Fermat point*, denoted by $\mathbf{x}^* = \mathbf{x}^*(A)$. The

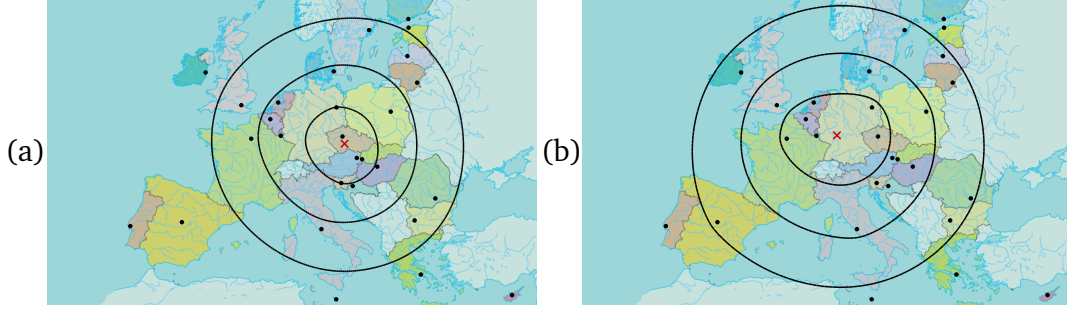


Figure 5.1. The Fermat point of the 28 EU-capitals (pre-Brexit), highlighted with (x), along with three 28-ellipses of different radii. (a) The foci (capitals) are unweighted. (b) Each focus has the weight of the country's population. The source of the map is <https://www.consilium.europa.eu>.

Fermat point is not unique if and only if A is collinear and n is even. We can check if A is collinear in $O(n)$ time, and in that case, the median, which is a Fermat point, can be found in $O(n \log n)$ time. So henceforth, we assume that A is not collinear. In that case φ is a strictly convex function Mello and dos Santos [2018]; Morrison [2010], and \mathbf{x}^* is unique.

We also consider the closely related problem of computing n -ellipses of A . For any $r > r^*(A)$, the level set of the Fermat distance function is $\varphi^{-1}(r) := \{\mathbf{x} \in \mathbb{R}^d : \varphi(\mathbf{x}) = r\}$. If $n = 1$, the level set is a sphere; and if $n = 2$ and $d = 2$, it is the classic ellipse. When A has n points, we call $\varphi^{-1}(r)$ an n -ellipsoid, or an n -ellipse if $d = 2$; hence the term *foci set*. From an application perspective, an n -ellipse of radius r can be viewed as a curve that bounds the candidate area for facility location Petrović et al. [2014], such that the total transportation cost to the demand points is at most r , as in Fig. 5.1.

The question of approximating the Fermat point is of great interest as its coordinates are roots of polynomials of degree exponential in n Bajaj [1988]. For any $\varepsilon > 0$, an ε -approximation $\tilde{\mathbf{x}}^*$ to the Fermat point \mathbf{x}^* can be interpreted in 3 ways:

- (A) **Approximate Fermat Point:** $\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| \leq \varepsilon$;
- (B) **Absolute Approximate Fermat Radius:** $\varphi(\tilde{\mathbf{x}}^*) \leq \varphi(\mathbf{x}^*) + \varepsilon$;
- (C) **Relative Approximate Fermat Radius:** $\varphi(\tilde{\mathbf{x}}^*) \leq (1 + \varepsilon)\varphi(\mathbf{x}^*)$.

Thus, we have three ε -approximation problems: (A), (B) and (C). Essentially (B) and (C) are approximations of the Fermat radius, while (A) is a direct approximation of the Fermat point. We consider approximations in the sense of (A); to

the best of our knowledge, only approximations in the sense of (B) and (C), have been considered before, see e.g., Bose et al. [2003]; Cohen et al. [2016]. Below, we show that (B) and (C) are easily *reduced* to (A) while the converse reductions are non-obvious (reductions in the sense of complexity theory).

In this work we introduce certified algorithms for approximating the Fermat point and n -ellipses, combining a subdivision approach with interval methods (cf. Lin and Yap [2011]; Ratschek and Rokne [1984]). The approach can be formalized in the framework of *soft predicates* Wang et al. [2015]. Our certified algorithms are fairly easy to implement, and are shown to have good performance experimentally.

Related Work. The problem we study has a long history, with numerous extensions and variations. Out of the 15 names found in the literature, see Hamacher and Drezner [2002], we call it *the Fermat point problem*. Other common names are the *Fermat-Weber problem* and the *Geometric median problem*. Apart from the Facility Location application introduced by Weber [1909], the problem is motivated by applications in diverse fields such as statistics and data mining where it is known as the *1-Median problem*, and is an instance of the k -median clustering technique Har-Peled and Mazumdar [2004].

For $d = 2, n = 3$, the problem was first stated by P. Fermat (1607 - 1665) and was solved by E. Torricelli (1608 - 1647) and Krarup and Vajda [1997] using a geometric construction. For $n = 4$, solutions were given by Fagnano [1775] and Cieslik [2013]. The first general method, for arbitrary n , is an iterative scheme proposed by Weiszfeld [1937] in 1937. It was later corrected and improved by Kuhn [1973] and Ostresh Jr [1978]; see Beck and Sabach [2015] for a review. The method is essentially a gradient descent iterative algorithm. It behaves quite well in practice and has only linear convergence, with guaranteed convergence from any starting point.

A plethora of approximation algorithms for the Fermat point, in the senses of (B) and (C), can be found in the literature using various methods. There are algorithms based on semidefinite programming Parrilo and Sturmfels [2003], interior point methods Cohen et al. [2016]; Xue and Ye [1997], sampling Badoiu et al. [2002]; Cohen et al. [2016], geometric data structures Bose et al. [2003] and coresets Har-Peled and Kushal [2007], among others Chin et al. [2013]; Feldman and Langberg [2011]. Moreover, special configurations of foci have been considered Bhattacharya [2011]; Cockayne and Melzak [1969], a continuous version of the problem Fekete et al. [2005], and also a generalized Fermat point of planar convex objects Abu-Affash and Katz [2009]; Carmi et al. [2005];

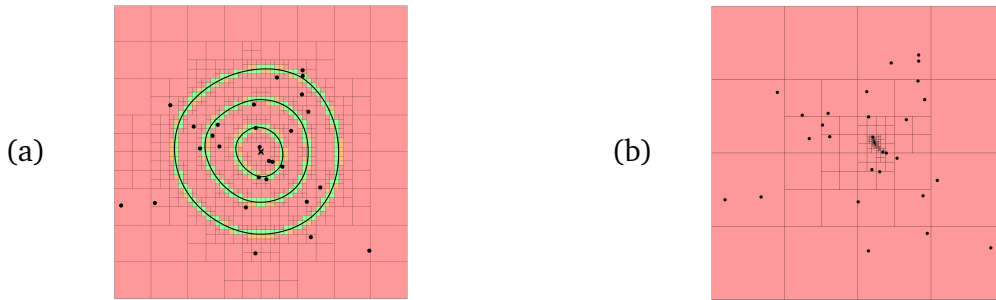


Figure 5.2. The resulting box subdivision of Fig. 5.1(a) for (a) the n -ellipses and (b) the Fermat point.

Dumitrescu et al. [2011].

The literature on n -ellipses is smaller but equally old: Nagy [1950] proved that n -ellipses are convex curves, calling them *egg curves*, and dating them back to [von Tschirnhaus, 1695, p. 183]. Further, he characterized the singular points of the n -ellipses as being either foci or the Fermat point. Another early work is by Sturm [1884]. Sekino [1999] showed that the Fermat distance function φ is C^∞ on $\mathbb{R}^2 \setminus A$. So, the n -ellipse is a piecewise smooth curve, as it may pass through several foci. Nie et al. [2008] showed that the polynomial equation defining the n -ellipses has algebraic degree exponential in n .

Our Contributions. We design, implement and experimentally evaluate algorithms for approximating the Fermat point of a given set of foci in \mathbb{R}^d . We also compute an ε -approximate n -ellipse; a problem not considered in computational literature before. These are the first certified algorithms Moore [1966]; Tucker [2011] for these problems. Our contributions are summarized as follows:

- We design two certified algorithm for the approximate Fermat point: one based on subdivision, the other based on Weiszfeld iteration Weiszfeld [1937].
- Our notion of ε -approximate Fermat point appears to be new; in contrast, several recent works focus on ε -approximation of the Fermat radius. The approximate Fermat radius can be reduced to approximate Fermat point; the converse reduction is unclear.
- Based on the *PV construction* Plantinga and Vegter [2004]; Lin and Yap [2011], we design an algorithm to compute a regular isotopic ε -approximation of an n -ellipse. We also augment the algorithm to compute simultaneous

contour plots of the distance function φ , resulting in a useful visualization tool (see Fig. 5.1).

- We implement and experimentally evaluate the performance of all our algorithms on different datasets in the plane, as a function of n and ε .

5.2 Preliminaries

Vector variables are written in bold font: thus $\mathbf{0}$ is the origin of \mathbb{R}^d and $\mathbf{x} = (x_1, \dots, x_d)$. For a differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, let $\partial_i f$ denote partial differentiation with respect to x_i . The *gradient* $\nabla f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ of f is given by the vector $\nabla f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x}))^T$ where $f_i = \partial_i f$. In general, the operator ∇ is partial, i.e., $\nabla f(\mathbf{x}_0)$ might not be defined at a point \mathbf{x}_0 . A point \mathbf{x}_0 is a *critical point* of f if $\nabla f(\mathbf{x}) = \mathbf{0}$ or $\nabla f(\mathbf{x})$ is undefined.

We consider analytic properties of a scalar function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, mainly from the viewpoint of convex analysis Mello and dos Santos [2018]; Nam [2013]. In our case, f is the Fermat distance function for some weighted set A . From an abstract perspective, the Fermat point problem (resp., n -ellipsoid problem) amounts to computing the critical points of the gradient of f (resp., computing the level sets of f). The Fermat point is the only critical point of ∇f in $\mathbb{R}^d \setminus A$, assuming A is non-collinear.

Most of the basic properties regarding the Fermat point are well-known and may be found in our references such as Kuhn [1973]; Mello and dos Santos [2018]; Nam [2013]; Ostresh Jr [1978]; Weiszfeld [1937]. To emphasize the foci set A , we explicitly write φ_A instead of φ . A focus $\mathbf{a} \in A$ is the Fermat point of A if and only if $\|\nabla \varphi_{A \setminus \mathbf{a}}(\mathbf{a})\| \leq w(\mathbf{a})$. Testing if the Fermat point \mathbf{x}^* is in A can be done in $O(n^2 d)$ time. If \mathbf{x}^* is not one of the foci, then $\nabla f(\mathbf{x}^*) = \mathbf{0}$, and the problem can be reduced to general finding real zeros of a square system of polynomial equations (e.g., Xu and Yap [2019]). However, the thrust of this paper is to develop direct methods that exploit the special properties of the Fermat problem.

We formally define the two main problems which we consider:

- APPROXIMATE FERMAT POINT: Given a weighted point set A in \mathbb{R}^d and $\varepsilon > 0$, compute a point $\tilde{\mathbf{x}}^*$ within ε distance to the Fermat point \mathbf{x}^* of A .
- APPROXIMATE ISOTOPIC n -ELLIPSES: Given $\varepsilon > 0$, a weighted point set A in \mathbb{R}^2 of size n and a radius $r > r^*(A)$, compute a closed polygonal

curve E that is ε -isotopic to $\varphi^{-1}(r)$, i.e., there exists an ambient isotopy¹ $\gamma : \mathbb{R}^2 \times [0, 1] \rightarrow \mathbb{R}^2$ with $\gamma(E, 1) = \varphi^{-1}(r)$ and for any point $\mathbf{a} \in \varphi^{-1}(r)$, the parametric curve $\gamma(\mathbf{a}, \cdot)$ has at most length ε . This implies a bound of ε on the Hausdorff distance between E and $\varphi^{-1}(r)$.

Approximation notions. We compare the three different notions of ε -approximation for the Fermat point. We reduce the approximation problem of notion (C) to (B), and (B) to (A). An ε -approximation $\tilde{\mathbf{x}}^*$ of \mathbf{x}^* in the sense $\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| \leq \varepsilon$ is also a $(W\varepsilon)$ -approximation in the sense $\varphi(\tilde{\mathbf{x}}^*) \leq \varphi(\mathbf{x}^*) + W\varepsilon$, which follows directly from the triangle inequality

$$\varphi(\tilde{\mathbf{x}}^*) = \sum_{\mathbf{a} \in A} w(\mathbf{a}) \|\tilde{\mathbf{x}}^* - \mathbf{a}\| \leq \sum_{\mathbf{a} \in A} w(\mathbf{a}) (\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| + \|\mathbf{x}^* - \mathbf{a}\|) = W\varepsilon + \varphi(\mathbf{x}^*).$$

An ε -approximation $\tilde{\mathbf{x}}^*$ of \mathbf{x}^* in the sense $\varphi(\tilde{\mathbf{x}}^*) \leq \varphi(\mathbf{x}^*) + \varepsilon$ is also a $\frac{2\varepsilon}{\varphi(g)}$ -approximation in the sense $\varphi(\tilde{\mathbf{x}}^*) \leq (1 + \frac{2\varepsilon}{\varphi(g)})\varphi(\mathbf{x}^*)$. The center of gravity g is a 2-approximation of the Fermat radius r^* (see Cohen et al. [2016]), i.e. $\varphi(\mathbf{x}^*) \geq \frac{1}{2}\varphi(g)$. Hence

$$\varphi(\tilde{\mathbf{x}}^*) \leq \varphi(\mathbf{x}^*) + \varepsilon = \left(1 + \frac{\varepsilon}{\varphi(\mathbf{x}^*)}\right)\varphi(\mathbf{x}^*) \leq \left(1 + \frac{2\varepsilon}{\varphi(g)}\right)\varphi(\mathbf{x}^*)$$

On the other hand, it is not clear how to derive an ε -approximation of type (A) if an approximation algorithm for type (B) and (C) is at hand, as the following 2 examples show.

Example 1: For any $\varepsilon > 0$ choose $c \leq \frac{\varepsilon}{2\sqrt{2}-2}$ and consider the weighted foci $\mathbf{a}_1 = (1, 0)$, $\mathbf{a}_2 = (0, 1)$, $\mathbf{a}_3 = (-1, 0)$, $\mathbf{a}_4 = (0, -1)$ with $w(\mathbf{a}_1) = w(\mathbf{a}_3) = 1$ and $w(\mathbf{a}_2) = w(\mathbf{a}_4) = c$ for which the Fermat point is $\mathbf{x}^* = (0, 0)$ for symmetry reasons, see Fig. 5.3(a). Point $p = (1, 0)$ is an ε -approximation of \mathbf{x}^* in the sense (B) and (C), but it has a distance of 1 to \mathbf{x}^* .

Example 2: For any $\varepsilon > 0$ we choose $h > 0$ small enough such that: $2\sqrt{4+h^2} + 2h \leq 4\sqrt{1+h^2} + \varepsilon$. Consider the foci $a_1 = (0, -h)$, $a_2 = (0, h)$, $a_3 = (2, -h)$, $a_4 = (2, h)$ with unit weights. The Fermat point is $\mathbf{x}^* = (1, 0)$ for symmetry reasons, see Fig. 5.3(b). Point $p = (2, 0)$ is an ε -approximation of \mathbf{x}^* in the sense (B) and (C), but it has a distance of 1 to \mathbf{x}^* .

Pure Newton: Lack of global convergence The algorithms that we developed put an emphasis on robustness. This is a real issue, which can go very wrong

¹That is, a continuous map $\gamma : \mathbb{R}^2 \times [0, 1] \rightarrow \mathbb{R}^2$ such that $\gamma_0 = \gamma(\cdot, 0)$ is the identity map, and, for all $t \in [0, 1]$, $\gamma_t = \gamma(\cdot, t)$ is a homeomorphism on \mathbb{R}^2 .

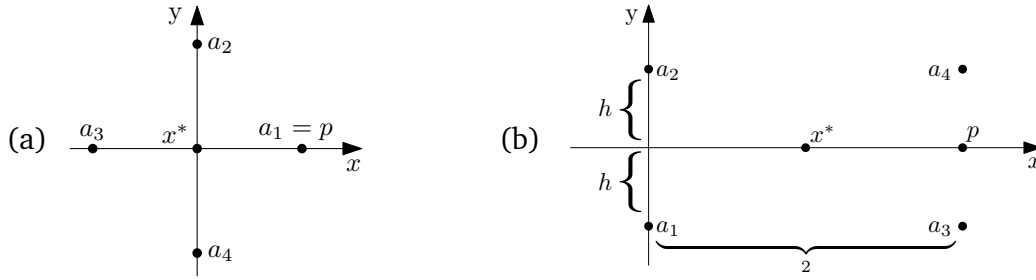


Figure 5.3. (a) Example that a good approximation of the Fermat point in sense (B) does not imply a good approximation in sense (A). (b) Analogous example for sense (C).

for naive algorithms as the following example shows. The point Newton method may fail to converge due to cycling between approximation points. Consider the following 10 foci in \mathbb{R}^3 , each with a weight of 1:

$$\begin{aligned} a_1 &= (0.38462, 0.58299, 0.25181) \\ a_2 &= (0.29044, 0.61709, 0.26528) \\ a_3 &= (0.82438, 0.98266, 0.73025) \\ a_4 &= (0.34388, 0.58407, 0.10777) \\ a_5 &= (0.90631, 0.87965, 0.81776) \\ a_6 &= (0.26073, 0.59436, 0.022513) \\ a_7 &= (0.42526, 0.31272, 0.16148) \\ a_8 &= (0.17877, 0.42289, 0.094229) \\ a_9 &= (0.59852, 0.47092, 0.69595) \\ a_{10} &= (0.69989, 0.63853, 0.033604) \end{aligned}$$

If we start with the center of mass $p_0 = \frac{1}{n} \sum_{a \in A} \mathbf{a}$ then for $f = \nabla \varphi$ the pure Newton method $p_{i+1} = p_i - J_f^{-1}(p_i) \cdot f(p_i)$ does not terminate. In particular, for big enough i the sequence keeps revisiting the following 4 points:

$$\begin{aligned} p_{4i} &= (0.40089, 0.58085, 0.23502) \\ p_{4i+1} &= (0.37393, 0.58077, 0.25124) \\ p_{4i+2} &= (0.43552, 0.58899, 0.24779) \\ p_{4i+3} &= (0.32493, 0.56753, 0.22338) \end{aligned}$$

Subdivision Paradigm. The subdivision algorithms presented here take as input an initial box $B_0 \subset \mathbb{R}^d$ and recursively split it. We organize the boxes in a *generalized quadtree* data structure Samet [1990]. A box can be specified by d intervals as $B = I_1 \times I_2 \times \cdots \times I_d$. Let m_B denote the *center* of B , r_B the *radius* of B (distance between m_B and a corner), and $\omega(B)$ the *width* of B (the maximum length of its defining intervals). The term $c \cdot B$ denotes the box with center m_B

and radius $c \cdot r_B$. The function SPLIT_1 takes a box B and returns 2^d congruent subboxes (*children*), one for each orthant. We use SPLIT_2 to indicate that we do two successive levels of SPLIT_1 operations (i.e., $1 + 2^d$ SPLIT_1 operations, resulting in $(2^d)^2 = 4^d$ leaves).

Soft Predicates. Let $\square\mathbb{R}^d$ denote the set of closed d -dimensional boxes (i.e., Cartesian products of intervals) in \mathbb{R}^d . Let P be a logical *predicate* on boxes, i.e., $P : \square\mathbb{R}^d \rightarrow \{\text{true}, \text{false}\}$. For example, the *Fermat point predicate* is given by $P_{\text{fp}}(B) = \text{true}$ if and only if $\mathbf{x}^* \in B$. Logical predicates are hard to implement, and thus, we may focus on *tests*, which are viewed as one-sided predicates. Formally, a test T looks like a predicate: $T : \square\mathbb{R}^d \rightarrow \{\text{success}, \text{failure}\}$ and it is always associated to some predicate P : call T a *test for predicate* P if $T(B) = \text{success}$ implies $P(B) = \text{true}$. However, we conclude nothing if $T(B) = \text{failure}$. Denote this relation by “ $T \Rightarrow P$ ”.

Soft predicates Wang et al. [2015] are an intermediate concept between a test and a predicate. Typically, they arise from a partial scalar function $f : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\uparrow\}$ where $f(\mathbf{x}) = \uparrow$ means $f(\mathbf{x})$ is not defined. We then define a partial *geometric predicate* P_f on boxes B as follows:

$$P_f(B) = \begin{cases} \uparrow & \text{if } \uparrow \in f(B), \\ +1 & \text{if } f(B) > 0, \\ -1 & \text{if } f(B) < 0, \\ 0 & \text{else.} \end{cases}$$

We can now derive various logical predicates P from P_f , by identifying the values in the set $\{-1, 0, +1, \uparrow\}$ with true or false. For instance, we call P an *exclusion predicate* if we associate the 0- and \uparrow -value with false and the other values with true. For the *inclusion* predicate, we associate the 0-value with true, others with false. For example, a test for the Fermat point predicate P_{fp} is an inclusion predicate based on the partial function $f(\mathbf{x}) = \sum_i (\partial_i f(\mathbf{x}))^2$; the function is partial because $f(\mathbf{x}) = \uparrow$ when \mathbf{x} is a focus point. Although our box predicates $P(B)$ are defined for full-dimensional boxes B , we can extend them to any point \mathbf{x} as follows: $P(\mathbf{x})$ has the logical value associated with the $\text{sign}(f(\mathbf{x})) \in \{\uparrow, +1, -1, 0\}$.

Definition 5.2.1. Let T be a test for a predicate P . We call T a *soft predicate* (or soft version of P) if it is convergent in this sense: if $(B_i : i = 0, 1, \dots)$ is a monotone sequence of boxes $B_{i+1} \subseteq B_i$ that converges to a point \mathbf{a} , then $P(\mathbf{a}) \equiv T(B_i)$ for i large enough.

Here, “ $P(\mathbf{a}) \equiv T(B_i)$ ” means $P(\mathbf{a}) = \text{true}$ if and only if $T(B_i) = \text{success}$. A soft version of $P(B)$ is usually denoted $\square P(B)$. We note that soft versions of exclusion predicates are generally easier to construct than inclusion predicates. The former can be achieved by numerical approximation, while the latter requires some deeper principle such as the Brouwer fixed point theorem Brouwer [1911].

Interval arithmetic. We construct soft predicates using functions of the form $F : \square\mathbb{R}^d \rightarrow \square(\mathbb{R} \cup \{-\infty, \infty\})$ that approximate the scalar function $f : D \rightarrow \mathbb{R}$ with $D \subset \mathbb{R}^d$.

Definition 5.2.2. Call F a soft version of f if it is

- i) conservative, i.e., for all $B \in \square\mathbb{R}^d$, $F(B)$ contains $f(B) := \{f(p) : p \in B \cap D\}$, and
- ii) convergent, i.e., if for monotone sequence $(B_i : i \geq 0)$ that converges to a point $\mathbf{a} \in D$, $\lim_{i \rightarrow \infty} \omega(F(B_i)) = 0$ holds.

We shall denote F by $\square f$ when F is a soft version of f . There are many ways to achieve $\square f$. For example, if f has an arithmetic expression E , we can simply evaluate E using interval arithmetic. More sophisticated methods may be needed for performance. The next lemma shows how $\square f$ leads to soft exclusion predicates based on f .

Lemma 5.2.3. If P is an exclusion predicate based on f , then the test $\square P(B) : 0 \notin \square f(B)$ is a soft version of P .

Below, we need a multivariate generalization, to the case where $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^m$, and the exclusion predicate $P(B)$ is $\mathbf{0} \notin \mathbf{f}(B)$. If $\square \mathbf{f} : \square\mathbb{R}^d \rightarrow \square\mathbb{R}^m$ is a soft version of \mathbf{f} , then a soft version of $P(B)$ is the given by the test $T(B) : \mathbf{0} \notin \square \mathbf{f}(B)$. If $\mathbf{f} = (f_1, \dots, f_m)$, then this reduces to $0 \notin \square f_i(B)$ for some $i = 1, \dots, m$.

5.3 Approximate Fermat points

We now present three approximation algorithms for the Fermat point \mathbf{x}^* . For simplicity, we assume in our algorithms that the Fermat point is not a focus, i.e. $\mathbf{x}^* \notin A$. This assumption can be easily checked in $O(n^2d)$ preprocessing time, or with a more elegant approach, in $O(nd)$ time during the execution of our subdivision algorithms.

The next theorem was already shown in Weiszfeld [1937], but we quickly sketch a proof again.

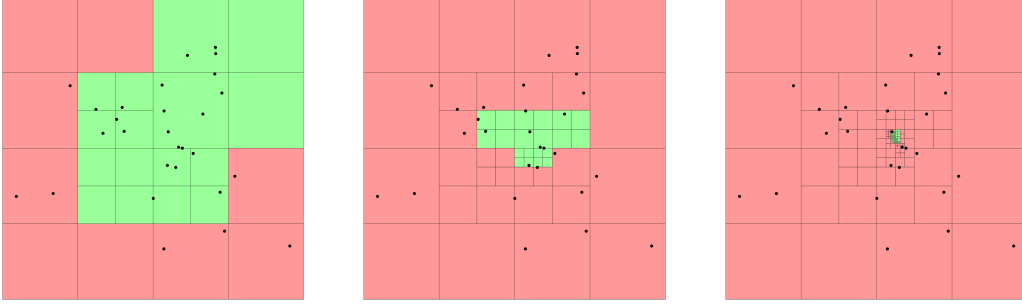


Figure 5.4. Different steps during the the execution of Algorithm 3. The dark red boxes cannot contain the Fermat point, whereas the light green boxes may contain it.

Theorem 5.3.1. *A focus \mathbf{a} is the Fermat point if and only if $\|\nabla\varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a})\| \leq w(\mathbf{a})$.*

Proof. We need to show that \mathbf{a} is the minimizer of the convex function φ if $\|\nabla\varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a})\| \leq w(\mathbf{a})$. Let $\mathbf{v} \in \mathbb{R}^2$ be any unit vector. The directional derivative of φ in direction \mathbf{v} at \mathbf{a} is

$$\begin{aligned} \lim_{h \rightarrow 0} \frac{\varphi(\mathbf{a} + h\mathbf{v}) - \varphi(\mathbf{a})}{h} &= \lim_{h \rightarrow 0} \frac{\varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a} + h\mathbf{v}) - \varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a})}{h} + \lim_{h \rightarrow 0} \frac{\varphi_{\{\mathbf{a}\}}(\mathbf{a} + h\mathbf{v}) - \varphi_{\{\mathbf{a}\}}(\mathbf{a})}{h} \\ &= \langle \mathbf{v}, \nabla\varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a}) \rangle + w(\mathbf{a}) \stackrel{\|\mathbf{v}\|=1}{\geq} -\|\nabla\varphi_{A\setminus\{\mathbf{a}\}}(\mathbf{a})\| + w(\mathbf{a}) \geq 0 \end{aligned}$$

Recall that $\langle \cdot, \cdot \rangle$ denotes the scalar product. This implies that starting from \mathbf{a} the function φ is non-decreasing in any direction. The minimum of the convex function φ therefore has to be \mathbf{a} . \square

5.3.1 Using the Subdivision Paradigm

The subdivision paradigm requires an initial box B_0 to start subdividing. If B_0 is not given, it is easy to find a box that contains \mathbf{x}^* , since \mathbf{x}^* lies in the convex hull of A Kuhn [1973]. We use a function INITIAL-BOX(A) which, in $O(nd)$ time, computes an axis-aligned bounding box with corners having the minimum and maximum x, y coordinates.

We define an exclusion and inclusion predicate based on the gradient function $\nabla\varphi$.

Definition 5.3.2. *Given a box B , the gradient exclusion predicate $C_0^\nabla(B)$ is defined by the condition $\mathbf{0} \notin \nabla\varphi(B)$. The gradient inclusion predicate $C_1^\nabla(B)$ is just the complement of $C_0^\nabla(B)$, that is $\mathbf{0} \in \nabla\varphi(B)$.*

Under our assumptions that $\mathbf{x}^* \notin A$, we have that $C_1^\nabla(B)$ holds if and only if $\mathbf{x}^* \in B$. We obtain a soft version of the exclusion predicate $C_0^\nabla(B)$ by replacing $\nabla\varphi$ in its definition with any soft version $\square\nabla\varphi$, see Lemma 5.2.3. But it is not so easy to get a soft version of $C_1^\nabla(B)$; we shall return to this when we treat the Newton operator below.

In Algorithm 3, using the exclusion predicate we discard boxes that are guaranteed not to contain \mathbf{x}^* (red in Fig. 5.4) and we split boxes that might contain \mathbf{x}^* (green in Fig. 5.4). While subdividing, we test whether we can already approximate \mathbf{x}^* well enough by putting a bounding box around all the boxes that are not excluded yet, using the following predicate.

Definition 5.3.3. *Given a set of boxes Q that contains the Fermat point, the stopping predicate $C^\varepsilon(Q)$ returns true, if and only if the minimum axis-aligned bounding box containing all boxes in Q has a radius at most ε .*

If C^ε returns true, then we can stop. Since the radius of the minimum bounding box is at most ε , the center of the box is an ε -approximate Fermat point $\tilde{\mathbf{x}}^*$.

Algorithm 3: Subdivision for the approximate Fermat point (*SUB*)

Input : Foci set A , constant $\varepsilon > 0$
Output : Point $\tilde{\mathbf{x}}^*$

- 1 $B_0 \leftarrow \text{INITIAL-BOX}(A)$; $Q \leftarrow \text{QUEUE}()$; $Q.\text{PUSH}(B_0)$;
- 2 **while not** $C^\varepsilon(Q)$ **do**
- 3 $B \leftarrow Q.\text{POP}()$;
- 4 **if not** $\square C_0^\nabla(B)$ **then**
- 5 $Q.\text{PUSH}(\text{SPLIT}_1(B))$;
- 6 **return** $\tilde{\mathbf{x}}^* \leftarrow \text{Center of the bounding box of } Q$;

Regarding the runtime of Algorithm 3, evaluating $\nabla\varphi$ and its soft version takes linear time in n . The subdivision approach induces an exponential dependency on d , as splitting a box creates 2^d many children. Further, a SPLIT_1 operation decreases the boxwidth by a factor of 2, therefore, Algorithm 3 cannot converge faster than linear in ε .

5.3.2 Enhancing the Subdivision Paradigm

In this section, we augment Algorithm 3 with a speed up based on a *Newton operator*, which will ensure eventual quadratic convergence.

The Newton operator. Newton-type algorithms have been considered in the past, usually independently of other methods, and thus suffer from lack of global convergence. Moreover, from a numerical viewpoint, such methods face the *precision-control problem*. Our algorithm integrates subdivision with the Newton operator (an old idea that goes back to Dekker [1967] in the 1960's), thus ensuring global convergence.

We want to find the Fermat point, i.e., the root of $f = \nabla\varphi$. Newton-type operators are well-studied in the interval literature, and they have the form $N = N_f : \square\mathbb{R}^d \rightarrow \square\mathbb{R}^d$. There are three well-known versions of N_f : the simplest version, from Moore [1966] and Nickel [1969], is

$$N(B) = m_B - J_f^{-1}(B) \cdot f(m_B),$$

where J_f is the Jacobian matrix of f . Since $f = \nabla\varphi$, this matrix is actually the Hessian of φ . The second version by Krawczyk [1969]; Shary [2004] is:

$$N(B) = m_B - K \cdot f(m_B) + (I - K \cdot f(B)) \cdot (B - m_B),$$

where K is any non-singular $d \times d$ matrix, usually chosen to be an approximation of $J_f^{-1}(m_B)$. The third version, by Hansen [2006]; Hansen and Sengupta [1981], can be viewed as a sophisticated implementation of the Moore-Nickel operator using an iteration reminiscent of the Gauss-Seidel algorithm, combined with preconditioning. Later we report on our implementation of the first two Newton operators. In general, the Newton operator $N(B)$ does not return a box even if B is a box; so we define $\square N(B)$ to be a box that contains $N(B)$. For simplicity, we assume that $\square N(B)$ is the smallest box containing $N(B)$ with the same aspect ratio as B .

The following three properties of Newton box operators are consequences of Brouwer's Fixed Point Theorem Brouwer [1911]; Nickel [1971]; Shary [2004]; Xu and Yap [2019]:

1. (Inclusion Property) If $N(B) \subseteq B$ then $\mathbf{x}^* \in N(B)$.
2. (Exclusion Property) If $N(B) \cap B = \emptyset$ then $\mathbf{x}^* \notin B$.
3. (Narrowing Operator) If $\mathbf{x}^* \in B$ then $\mathbf{x}^* \in N(B)$.

Based on these properties, we can define two tests and an operator:

Definition 5.3.4. *Newton tests for gradient exclusion/inclusion predicates (below we explain why we use $2B$ instead of B):*

- **Newton exclusion test:**
 $T_0^N(B) = \text{success}$ iff $N(2B) \cap B = \emptyset$. Thus $T_0^N(B) \Rightarrow C_0^\nabla(B)$.
- **Newton inclusion test:**
 $T_1^N(B) = \text{success}$ iff $N(2B) \subseteq 2B$. Thus $T_1^N(B) \Rightarrow C_1^\nabla(2B)$.
- **Newton narrowing operator:**
 $N_\cap(B)$ returns $B \cap N(2B)$.

Note that the Newton tests $T_0^N(B)$ and $T_1^N(B)$ are defined using the exact Newton operator $N(B)$. If we replace it by a soft version $\square N(B)$ in these definitions, they remain as inclusion/exclusion tests for $C_0^\nabla(B)$ and $C_1^\nabla(B)$; we denote them by $\square C_0^\nabla(B)$ and $\square C_1^\nabla(B)$.

To compute $\square N(B)$, we use standard interval arithmetic to evaluate the Newton operators. We already noted that if $N(B) \subseteq B$, then $\mathbf{x}^* \in N(B)$. But if \mathbf{x}^* is on the boundary of B , then $\square N(B) \subseteq B$ might not hold, and this issue persists even after splitting B . We circumvent this problem by using $2B$ instead of B in the definition of $T_1^N(B)$.

We enhance Algorithm 3 by the soft inclusion predicate $\square T_1^N(B)$, as sketched in Algorithm 4. If $\square T_1^N(B)$ succeeds, we conclude that \mathbf{x}^* is contained in $\square N(2B)$. In that case, we can discard all other boxes and initialize a new queue Q on $\square N(2B)$. In subsequent calls to $\square T_1^N(B')$ for $B' \in Q$, we conclude that $\mathbf{x}^* \in 2B'$. But to ensure that $w(2B') < w(B)$ (to avoid an infinite loop), we initialize the queue Q with the 4^d boxes of $\text{SPLIT}_2(\square N(2B))$.

Algorithm 4: Enhanced subdivision for the approximate Fermat point
(*ESUB*)

As in Algorithm 3 but replace line 5 with the following:

```

5.1 | if  $\square T_1^N(B)$  then
5.2 |    $Q \leftarrow \text{QUEUE}()$ ;           // initialize a new queue
5.3 |    $Q.\text{PUSH}(\text{SPLIT}_2(\square N(2B)))$ ; // 2 split operations
5.4 | else
5.5 |    $Q.\text{PUSH}(\text{SPLIT}_1(B))$ ;

```

With respect to the runtime of Algorithm 4, we observe that once the soft Newton inclusion predicate succeeds, then it will also do so for an initial box of

the new queue. This, essentially, divides the algorithm into two phases. The first phase can be basically seen as Algorithm 3. In the second phase, the Newton test guarantees quadratic convergence in ε . Getting into the second phase depends on the configuration of the foci set but not on ε , hence, our approach is of particular interest for small values of ε .

The termination of both subdivision algorithms follows from the soft gradient exclusion predicate being convergent. The algorithms terminate once the predicate $C^\varepsilon(Q)$ succeeds, yielding an ε -approximate Fermat point, so we summarize as follows.

Theorem 5.3.5. *Both Algorithms 3 and 4 terminate and return an ε -approximate Fermat point.*

Proof. Assume Algorithm 3 does not terminate. Then the algorithm produces an infinite monotone sequence of boxes $(B_i : i = 0, 1, \dots)$ with $B_{i+1} \subseteq B_i$ that converges to a point \mathbf{p} , which is different from the Fermat point \mathbf{x}^* . For large enough i the soft gradient exclusion predicate will succeed, i.e. $\square C_0^\nabla(B_i) = \text{success}$, because it is a soft version of C_0^∇ . But this contradicts the assumption, that the algorithm would split box B_i , thus Algorithm 3 terminates. For the same reasoning also Algorithm 4 terminates, as the Newton inclusion predicate can only succeed near the Fermat point.

Both Algorithms 3 and Algorithm 4 terminate when the stopping criterion C^ε succeeds, which guarantees that the output of the algorithm is an ε -approximation of the Fermat point. \square

5.3.3 Certifying the Weiszfeld method

Weiszfeld's iterative method Kuhn [1973]; Ostresh Jr [1978]; Weiszfeld [1937] describes a sequence \mathbf{p}_i ($i = 0, 1, \dots$) of points that converges to the Fermat point \mathbf{x}^* , starting from any initial \mathbf{p}_0 . The recurrence relation is $\mathbf{p}_{i+1} = T(\mathbf{p}_i)$, where $T(\mathbf{x})$ is defined by

$$T(\mathbf{x}) = \frac{\sum_{\mathbf{a} \in A, \mathbf{a} \neq \mathbf{x}} w(\mathbf{a}) \frac{\mathbf{a}}{\|\mathbf{x} - \mathbf{a}\|}}{\sum_{\mathbf{a} \in A, \mathbf{a} \neq \mathbf{x}} w(\mathbf{a}) \frac{1}{\|\mathbf{x} - \mathbf{a}\|}}.$$

Note that when \mathbf{x} is a focus, then $T(\mathbf{x})$ depends just on all other foci.

This simple iterative method is widely used, and although it converges, it does not solve our ε -approximation problem as we do not know when to stop. To see that this is a real issue, consider the example explained next and illustrated in Fig. 5.5.

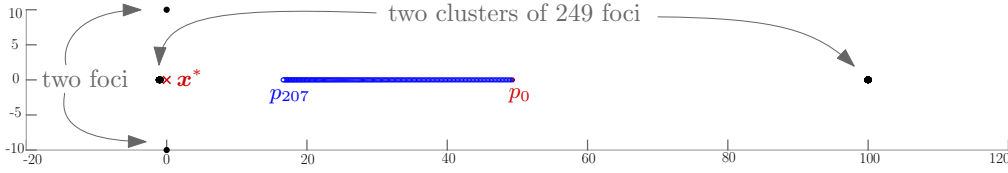


Figure 5.5. An example with 500 foci, showing that Weiszfeld's scheme does not solve the ε -approximation problem. The scheme stopped when $\|\mathbf{p}_{i-1} - \mathbf{p}_i\| \leq 1/10$, after 207 steps (blue points). The distance $\|\mathbf{x}^* - \mathbf{p}_{207}\|$ can be arbitrarily big ($\|\mathbf{x}^* - \mathbf{p}_{207}\| > 15$ in this case).

Since we seek an ε -approximation of the Fermat point, a simple heuristic for stopping the Weiszfeld sequence $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots$ is to stop at \mathbf{p}_i when $\|\mathbf{p}_{i-1} - \mathbf{p}_i\| < \varepsilon$. The example in Figure 5.5 shows 500 foci (black points), two of which have coordinates $(0, 10)$ and $(0, -10)$ and the others are two very dense clusters of 249 foci around $(0, 0)$ and $(100, 0)$, respectively. This configuration yields a Fermat point $\mathbf{x}^* \simeq (0.019, 0)$ (red 'x') which is very far away from the start point (red point) which is chosen as the center of mass $\simeq (49.3, 0)$. Setting $\varepsilon = 1/10$ Weiszfeld's scheme (blue points) was stopped after 207 steps because our simple stopping heuristic is satisfied when the step length at iteration $i = 207$ is smaller than $\varepsilon = 1/10$, i.e. $\|\mathbf{p}_{206} - \mathbf{p}_{207}\| \leq 1/10$. However, the distance $\|\mathbf{x}^* - \mathbf{p}_{207}\| > 15$ is still very big. Thus, a more advanced technique is needed to decide when to stop.

We augment the Weiszfeld iteration by adding Newton tests during the computation, turning it into an ε -approximation algorithm. While at the i -th iteration, we define a small box B with point \mathbf{p}_i as center, and map it to the box $\square N(B)$ using the Newton operator; see Fig. 5.6. If $\square N(B) \subseteq B$, then the Fermat point \mathbf{x}^* lies in $\square N(B)$. On the contrary, if $\square N(B) \not\subseteq B$ we move on to the next point \mathbf{p}_{i+1} and adjust the box size as follows.

If $\frac{B}{10} \cap \square N(\frac{B}{10}) = \emptyset$, then the box $\frac{B}{10}$ does not contain \mathbf{x}^* and we therefore expand B by a factor of 10. If $\frac{B}{10} \cap \square N(\frac{B}{10}) \neq \emptyset$, then there might be a focus in box $\frac{B}{10}$, which hinders $\square N(B) \subseteq B$ to succeed. In that case we shrink B by a factor of 10. If a focus is not in $\frac{B}{10}$, shrinking B does not effect the algorithm negatively, as B can expand again.

Using these tests we augment the point sequence scheme, sketched in Algorithm 5, with the property that if the Newton test evaluates to true, then we are guaranteed an ε -approximation of \mathbf{x}^* . As a starting point, we choose the *center of mass* \mathbf{p}_0 of A , i.e., $\mathbf{p}_0 = \frac{1}{W} \sum_{\mathbf{a} \in A} w(\mathbf{a}) \mathbf{a}$.

With respect to the runtime, the point sequence $T(\mathbf{x})$ converges linearly in

ε towards \mathbf{x}^* Katz [1974] but in order for Algorithm 5 to terminate the test $\square N(B) \subseteq B$ must succeed. Similar to other Newton operators, $\square N(B) \subseteq B$ succeeds for boxes in a neighborhood surrounding \mathbf{x}^* . This neighborhood depends only on the configuration of A but not on ε . Further, evaluating $T(\mathbf{x})$ and $\square N(B)$ can be done in $O(nd^2)$ time. We conclude as follows.

Theorem 5.3.6. *Algorithm 5 terminates and returns an ε -approximate Fermat point.*

Algorithm 5: Certified Weiszfeld for the approximate Fermat point (CW)

Input: Foci set A , constant $\varepsilon > 0$ **Output:** Point $\tilde{\mathbf{x}}^*$

```

1  $\mathbf{p} \leftarrow \mathbf{p}_0$ ;  $l \leftarrow \varepsilon$ ;
2 while TRUE do
3    $B \leftarrow \text{Box } B(m_B = \mathbf{p}, \omega(B) = l)$ ;
4   if  $\square N(B) \subseteq B$  then                                     // Fig. 5.6(a)
5      $\text{return } \tilde{\mathbf{x}}^* \leftarrow \mathbf{p}$ ;
6   else if  $\square N(\frac{B}{10}) \cap \frac{B}{10} = \emptyset$  then             // Fig. 5.6(b)
7      $l \leftarrow \min\{10 \cdot l, \varepsilon\}$ ;
8   else                                                       // Fig. 5.6(c)
9      $l \leftarrow \frac{1}{10} \cdot l$ ;
10   $\mathbf{p} \leftarrow T(\mathbf{p})$ ;

```

For simplicity, we had assumed for Algorithms 3, 4 and 5 that the Fermat point is not a focus. Note that this assumption can be checked in advance by evaluating $\|\nabla \varphi_{A \setminus \{a\}}(\mathbf{a})\| \leq w(\mathbf{a})$ for each focus \mathbf{a} , see Theorem 5.3.1, which would take (n^2d) time. We will explain two reasons, why that quadratic testing time in n can be avoided for both subdivision algorithms.

We added this assumption, because the Newton tests cannot succeed for a box B if B contains the Fermat point. This is because $\square \nabla^2 \varphi(B)$ has the interval $[\|m_B - \mathbf{a}\| - r, \|m_B - \mathbf{a}\| + r]$ in its denominator, which contains 0 if $\mathbf{a} \in B$. Hence the box $N(B)$ covers the whole space if a focus is in B .

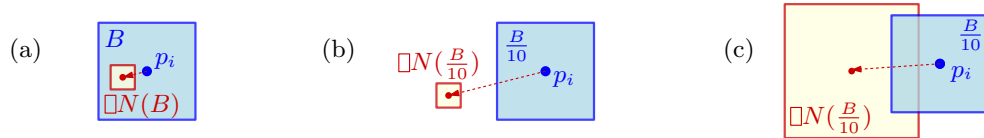


Figure 5.6. The case analysis of Algorithm 5. (a) $\square N(B) \subseteq B$, (b) $\square N(\frac{B}{10}) \cap \frac{B}{10} = \emptyset$, and (c) $\square N(\frac{B}{10}) \cap \frac{B}{10} \neq \emptyset$.

Instead of checking the assumption, one can run Algorithm 3 and 4 anyway and rely only on the soft gradient exclusion predicate. There is a more elegant solution for this problem, described next. Instead of testing all foci in the beginning, if one of them is the Fermat point, this can be done during the subdivision process. We keep track of the number of foci, which are contained in non-discarded boxes. If that number falls below a constant, then we test these few constantly many remaining foci for being the Fermat point. That can now be done in $O(nd)$ time.

5.4 Approximating n -ellipses

In this section, we describe an algorithm to construct approximate n -ellipses, based on the subdivision paradigm. Throughout this work we maintain the subdivision *smooth*, i.e., the width of any two adjacent boxes, which are leaves of the quadtree, may differ at most by a factor of 2. Maintaining smoothness is easy to implement and has amortized $O(1)$ cost per operation Bennett and Yap [2017]. Without maintaining smoothness, the amortized cost can be $\Omega(\log n)$ Bennett and Yap [2017].

The Plantinga and Vegter (PV) construction Plantinga and Vegter [2004]; Lin and Yap [2011]; Lin et al. [2012] approximates the zero set of a function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ where $d \in \{2, 3\}$. Assuming that $S = F^{-1}(0)$ is regular, i.e., the gradient ∇F is non-zero at every point of S , this approximation is isotopic to S . Our goal is to use this construction to approximate the n -ellipse defined by $F(p) := \varphi(p) - r$ with $r > r^*$. For simplicity, we assume all boxes are square; for the construction to succeed, we only need an aspect ratio $\leq \sqrt{2}$ (see Lin and Yap [2011]). We use the notation $\langle \cdot, \cdot \rangle$ for the scalar product. The following are the key predicates and tests in the PV construction of the n -ellipse $F^{-1}(0)$.

Definition 5.4.1. Fix $F(p) = \varphi(p) - r$. Let B be a square box.

1. The fundamental box predicate is the inclusion predicate $C_1^F(B) : 0 \in F(B)$, and its complement, the exclusion predicate $C_0^F(B) : 0 \notin F(B)$.
2. The (corner) inclusion test $T_{\text{cor}}(B) = \text{success}$ iff F , when evaluated at the corners of B , admits both negative and positive values. Clearly, $T_{\text{cor}}(B)$ is a test for $C_1^F(B)$. (There is a standard PV trick whereby any 0-value can be arbitrarily made positive.)
3. The normal variation predicate $C_{\text{nv}}(B)$ is defined by the condition $\langle \nabla F(B), \nabla F(B) \rangle > 0$.

We obtain the soft versions $\square C_0^F(B)$ and $\square C_{\text{nv}}(B)$ by the usual device of replacing $F(B)$ in the definition of the predicates by a soft version $\square F(B)$. But for the inclusion predicate $C_1^F(B)$ we have no soft version. Instead, the corner test $T_{\text{cor}}(B)$ is a test for $C_1^F(B)$. To supplement the corner test, we need the normal variation predicate $C_{\text{nv}}(B)$. This predicate is equivalent to the condition that the angle between the gradient of any two points in B is at most 90° . It implies that the n -ellipse is monotone in either x - or y -direction within the box. In Fig. 5.7, boxes are: **red** if they pass the $\square C_0^F(B)$ test, **green** if they pass both $\square C_{\text{nv}}$ and T_{cor} , **orange** if they pass only $\square C_{\text{nv}}$, and **gray** otherwise. Note that orange boxes may, or may not, contain parts of the approximate n -ellipse.

An n -ellipse is not regular if it passes through some focus Sekino [1999]; in that case a direct PV construction is not possible. We develop a variation, sketched in Algorithm 6, where we simultaneously subdivide boxes and construct pieces of the n -ellipse *on the fly*, instead of doing that in the end. Further, boxes in which the n -ellipse may not be regular are treated differently. During the subdivision part of the algorithm, we classify boxes in three categories:

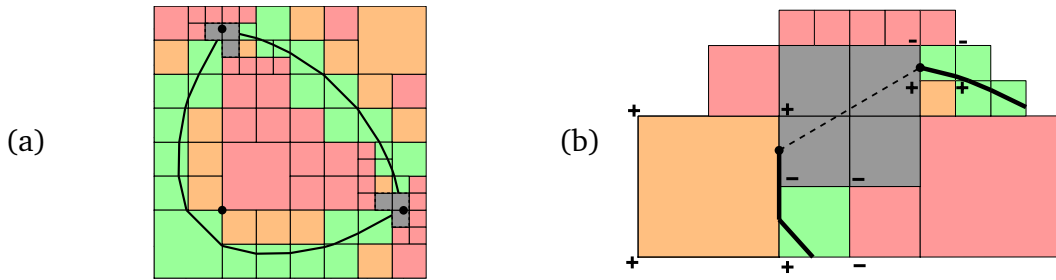


Figure 5.7. (a) A 3-ellipse passing through two foci. Components of gray boxes (temporarily) surround the foci. (b) If a gray component satisfies (B1) - (B3) the two ingoing edges are connected with an edge (shown dashed).

1. Boxes which satisfy $\square C_0^F(B)$ (**red**): These do not contain any piece of the n -ellipse, so they do not need to be further considered and are discarded.
2. Boxes which satisfy $\square C_{\text{nv}}$ and have width smaller than $\varepsilon/2$ (**green** or **orange**): We immediately draw edges in each of these boxes, in contrast to the normal PV construction. Note that at a later stage of the algorithm it might happen that we split one of B 's neighboring boxes. In that case we need to take into account the sign of F at the new vertex on B 's boundary. If necessary, the edges in box B then need to be updated.
3. The remaining boxes (**gray**): Such boxes occur near foci and need more careful attention, as we cannot apply the standard PV construction. Instead,

given a set of gray boxes we first distinguish them in connected components, using a DFS algorithm. Then, for each connected component of gray boxes K_i , we check if a set of conditions is satisfied:

- (B1) K_i contains exactly one focus.
- (B2) There are exactly two PV-edges leading to K_i .
- (B3) The distance between any two corners of the boxes in K_i is at most $\varepsilon/2$.

If K_i satisfies all (B1) - (B3), then we connect the 2 PV-edges leading to K_i by a line segment and discard boxes of K_i , see Fig. 5.7(b). Otherwise, the children of the boxes of K_i are put back in Q for further classification.

Algorithm 6: Approximating an n -ellipse

Input: Foci set A , radius r , constant ε , box B_0	Output: Curve E
--	--------------------------

```

1  $Q \leftarrow \text{QUEUE}(); \quad Q.\text{PUSH}(B_0);$ 
2 while  $Q \neq \emptyset$  do
3    $Q_{\text{new}} \leftarrow \text{QUEUE}();$ 
4   while  $Q \neq \emptyset$  do
5      $B \leftarrow Q.\text{POP}();$ 
6     if not  $\square C_0^F(B)$  then // exclude red
7       if  $\square C_{nv}(B)$  and  $\omega(B) < \varepsilon/2$  then // green or orange
8          $E_{\cap B} \leftarrow \text{ONLINE-PV}(B);$ 
9       else // gray
10         $Q_{\text{new}}.\text{PUSH}(\text{SPLIT}_4(B));$ 
11    $Q \leftarrow \text{CONNECTED-COMPONENTS-ANALYSIS}(Q_{\text{new}})$ 
12 return  $E;$ 

```

By controlling the size of the boxes containing parts of the output curve, and by the modification the PV construction we prove the following.

Theorem 5.4.2. *Algorithm 6 returns an isotopic ε -approximation of the n -ellipse $F^{-1}(0)$.*

Proof. The standard PV construction terminates for regular curves $S = F^{-1}(0)$. This implies that boxes of type (A3) can only survive in the neighborhood of foci. As time passes those neighborhoods become smaller and the neighborhoods of 2 different foci will become disjoint. That means that, eventually, properties (B1) -

(B3) will be satisfied for each component and no box will be put back to queue Q in line 11 of Algorithm 6.

The property that the output is a regular isotopic approximation of the n -ellipse is inherited from the PV-construction of regular curves. In the following we show that it is also an ε -approximation of the k -ellipse.

Let $S = F^{-1}(0)$ and S^* its approximation derived by Algorithm 6. We prove that the distance from any point on S^* to S is at most ε . A green or orange box B contains an edge of S^* only if F admits different signs when evaluated at corners on B 's boundary. In that case also the k -ellipse has to pass through B . The box radius of B is smaller than $\varepsilon/2$ and therefore any point on S^* in B has at most ε distance to S . Let p be a point on S^* in a gray box of component K . The component K has two ingoing edges and in particular two points on its boundary, which are on S , see Fig. 5.7(b). Therefore the distance from p to S can be bounded by the diameter of K , which is smaller than ε . None of the red boxes contains a part of S^* .

Finally we prove that the distance from any point on S to S^* is at most ε . All the boxes, which might contain parts of S satisfy the C_1 predicate (green and orange) or are part of a small component of gray boxes. If a box satisfies the C_1 predicate but the function F has the same sign at all its corners, then the curve S might possibly enter the box but also has to leave the box on the same side of B Plantinga and Vegter [2004] and any neighboring box on that side has different signs for F on its corners. Let p be a point on S in box B and let B_1 and B_2 be the next boxes which are reached by walking from p along S in both directions. Note that B_1 and B_2 might be the same box. If B is a gray box of component K , then the distance from p to S^* can be bounded by the distance between p and the edge of S^* in K . This distance is bounded by the diameter of K which is less than $\varepsilon/2$. If B is a green or orange box, then it satisfies the C_1 predicate and box B_1 and B_2 have different signs at their 4 corners. If B_1 or B_2 are green or orange then the approximation S^* passes through them and p is close enough to S^* . If both B_1 and B_2 are gray, then the edge of S^* through their components is close to p . Finally, B cannot be a red box by definition. \square

Interpolating edges. The PV construction creates edges within a box B , which start and end at midpoints of box edges. One can derive a nicer-looking approximation by using linear interpolation on the box edges by taking into account the value of F at B 's corners.

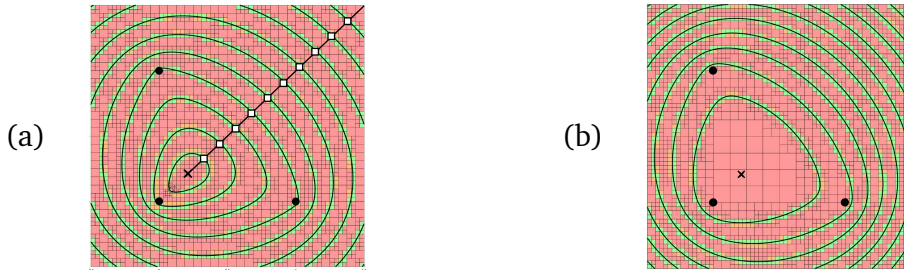


Figure 5.8. Two different 3-elliptic contour plots with 10 contour lines, having the same set of foci. (a) Using radii of equidistant points. (b) Using equidistant radii.

Contour Plotting. As an application, we can use the above technique in order to produce a topologically correct, ε -approximate and visually nice n -elliptic contour plot. To do so, we first adapt our algorithm in order to simultaneously plot several n -ellipses inside a bounding box, corresponding to the same foci but with different radii. Each n -ellipse is a *contour line*, and we describe how to plot them *visually nice*, i.e., the contour lines are roughly equally distributed in space. See Fig. 5.8 for two different approaches and their visualization effect.

5.5 Experiments

We implemented our algorithms for \mathbb{R}^2 and conducted a series of experiments. Our current software is written in MATLAB (version R2018b), taking advantage of its graphics ability. The numerical accuracy is therefore IEEE numerical precision. The platform used was MacOS Big Sur v11.2.3, with 2.5 GHz Quad-Core Intel Core i7 and 16 GB 1600MHz DDR3.

Following, we report on our experiments, discussing some notable points one by one. We evaluated our algorithms on both synthetic and real-world datasets. For all algorithms approximating the Fermat point we chose a time limit of 600 seconds. Moreover, for most experiments we executed 10 different instances for completeness. In the illustrated charts, the curves pass through the mean of the 10 running times, and additionally we also marked the minimum and maximum running times. All axes in the charts are of logarithmic scale.

Datasets. We mainly experimented with two different types of synthetic datasets, namely UNIF-1 and UNIF-2. In UNIF-1 the n foci are sampled uniformly from a disk of radius 1. In UNIF-2 again the n foci are sampled uniformly from

a disk of radius 1 and then $n/2$ foci are translated by a vector $(10, 10)$, see Fig. 5.9(a) and Fig. 5.9(b). Despite their similarity, the two datasets present strong differences. As we later see, UNIF-2 is significantly more difficult to solve in comparison to UNIF-1, and further UNIF-1 resembles nicely real-world datasets. The foci of UNIF-2 lie almost all on a common line, which implies that there are many points for which the gradient is close to 0. This makes it difficult to find the actual Fermat point, for which the gradient is exactly 0. We experimented with more types of synthetic datasets, such as points in convex position, vertices of a regular n -gon, clusters of points, but we do not report on these results, as they are similar to UNIF-1 or UNIF-2.

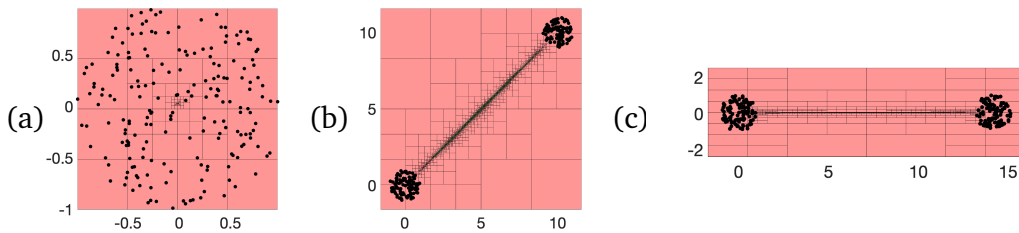


Figure 5.9. A box subdivision for $n = 200$ foci: (a) Unif-1, (b) Unif-2 and (c) Unif-2 after PCA.

Newton operators. Adding a Newton operator to the subdivision process drastically improves the running time. We compared Algorithm 3 with two versions of Algorithm 4, where we once use the Newton operator based on Moore and Nickel and also the operator by Krawczyk. The results for various values of n and ε on both UNIF-1 and UNIF-2 are summarized in Fig. 5.10. Note that Algorithm 4 initially needs to perform simple splitting operations until at some point the Newton test succeeds the first time. After that the algorithm converges quadratically in ε , which explains why the running time of both versions almost do not increase for decreasing ε . Even though the operator by Krawczyk returns a smaller box $\square N(B)$, i.e. it is more precise, than Moore and Nickel, it performs slower for UNIF-1 as evaluating the operator takes more time. We conclude that using a Newton operator speeds up the computations, and we use the one of by Moore and Nickel in Algorithm 4.

Principal component analysis. Foci sets like UNIF-2 are challenging as all foci are close to a common line. In this case, the subdivision algorithms can be slow because there are many boxes for which the gradient $\nabla\varphi$ is close to 0. Our

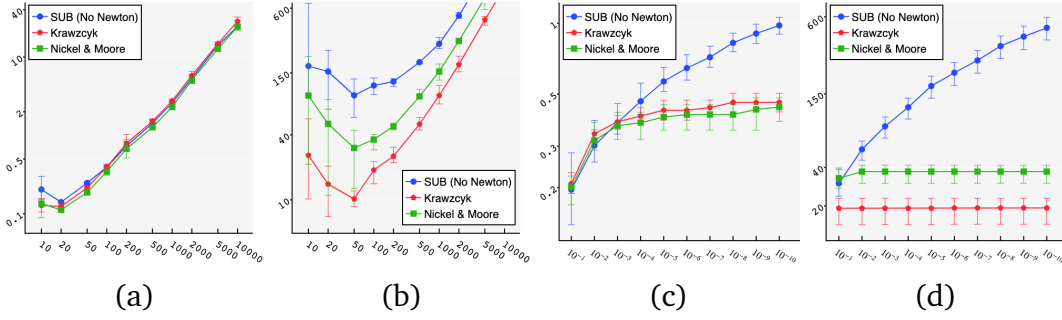


Figure 5.10. A comparison of Algorithm 3 (\bullet SUB), Algorithm 4 with the Krawczyk Newton operator (\diamond Krawczyk), and Algorithm 4 with the Nickel and Moore Newton operator (\blacksquare Nickel & Moore). (a),(b) Time as a function of n , with $\epsilon = 10^{-4}$. (c),(d) Time as a function of ϵ with $n = 100$. (a),(c) Unif-1 datasets. (b),(d) Unif-2 datasets.

approach to tackle this problem is to use subdivision with rectangular boxes. In a preprocessing step we do a *principal component analysis* (PCA) of the foci as heuristic. Then, we rotate the coordinate system such that the x -direction is the first principal component. In the box subdivision we use rectangular boxes with long x -width, see Fig. 5.9(c). Observe in the following table, that for well distributed foci sets like UNIF-1, using the PCA adds only a small overhead to the total running time.

$\epsilon = 10^{-3}, n =$	10	100	1000	10000	$n = 100, \epsilon =$	10^{-1}	10^{-3}	10^{-5}	10^{-7}
without PCA	0.12	0.31	2.33	23.4	without PCA	0.20	0.30	0.33	0.34
with PCA	0.10	0.30	2.30	23.9	with PCA	0.18	0.30	0.33	0.35

On the contrary, for sets like UNIF-2, adding the PCA decreases drastically the running time, as shown next. Hence, the PCA preprocessing is a useful addition to Algorithm 4, which we will use also in the following experiments.

$\epsilon = 10^{-3}, n =$	10	100	1000	10000	$n = 100, \epsilon =$	10^{-1}	10^{-3}	10^{-5}	10^{-7}
without PCA	90.7	48.5	170	timeout	without PCA	37.1	49.2	49.2	49.5
with PCA	0.15	0.40	3.21	32.7	with PCA	0.36	0.40	0.42	0.43

Real Datasets. Inspired by the applications in facility location we chose to experiment with instances of the well-known *Traveling Salesman Person Library* Reinelt [1991] or TSPLIB. The foci correspond mostly to location of cities in different areas around the world, see Fig. 5.11.

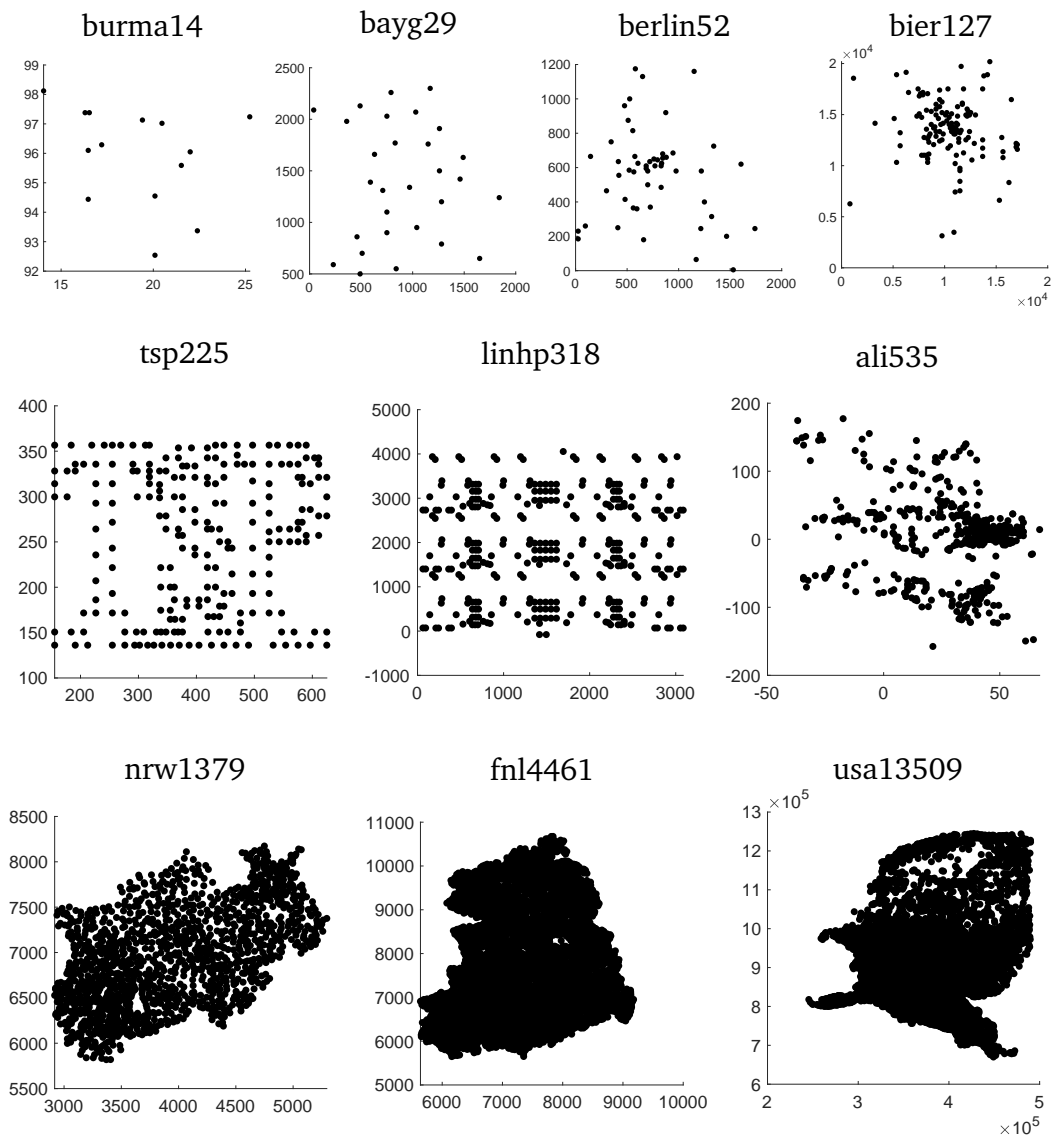


Figure 5.11. Foci sets of TSPLib used in our experiments. burma14: 14 cities in Burma, bayg29: 29 cities in Bavaria, berlin52: 52 locations in Berlin, bier127: 127 beer gardens in the Augsburg area (Germany), tsp225: writing of TSP with 225 points, linhp318: 318 cities, ali535: 535 airports around the globe, nrw1379: Nordrhein-Westfalen (Germany), fnl4461: the five Federal States of Germany (ex-GDR territory), usa13509: cities in the continental US with at least 500 population

It appears that real-world instances show a similar behavior to UNIF-1 datasets; so we infer that UNIF-1 are realistic datasets for the evaluation of different algorithms. In our experiments, to verify that for each TSPLIB dataset we created

an additional foci set, where we uniformly sampled the same number of foci in the axis-aligned bounding box. As ε we chose 10^{-6} times the width of the corresponding bounding box. These experiments are illustrated in Fig. 5.14(a), and the similarity of the running time for the two datasets is obvious.

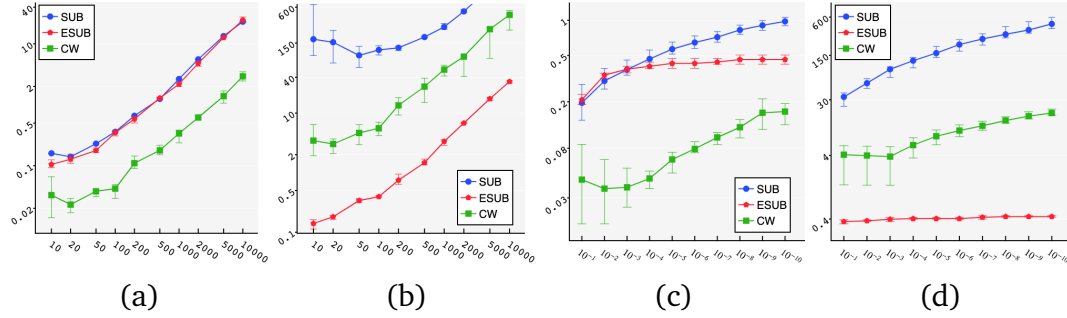


Figure 5.12. An overall comparison of Algorithm 3 (\bullet SUB), Algorithm 4 with the PCA (\square ESUB), and Algorithm 5 (\blacksquare CW). (a),(b) Time as a function of n , with $\varepsilon = 10^{-4}$. (c),(d) Time as a function of ε with $n = 100$. (a),(c) Unif-1 datasets. (b),(d) Unif-2 datasets.

Comparison with the Interval method We compared our Algorithm 3 with a naive approach, called *interval method*. It is based on the fact that if given two boxes B_1 and B_2 , such that the intervals $\square\varphi(B_1)$ and $\square\varphi(B_2)$ are disjoint, then the box with bigger function values cannot contain the Fermat point. The interval method is a subdivision algorithm like Algorithm 3, where at any time we keep track of the smallest upper bound b of intervals $\square\varphi(B)$, for boxes B visited so far. The interval method replaces the soft gradient exclusion predicate in line 4 of Algorithm 3 by the other soft exclusion predicate $b < \square\varphi(B)$. We compared these two methods for different values of n and ε using the data sets UNIF-1. The results are summarized in the next two tables. In all tests the soft gradient exclusion predicate performed much better. Note, that for boxes B near the Fermat point the value $\square\varphi(B)$ is very similar. Hence, the interval method needs to do many splitting operations for small ε and work with very high internal precision. This explains, why that naive method did not terminate within 600 sec for $n = 100$ and $\varepsilon = 10^{-7}$.

$\varepsilon = 10^{-3}, n =$	10	100	1000	10000	$n = 100, \varepsilon =$	10^{-1}	10^{-3}	10^{-5}	10^{-7}
Interval method	0.99	1.72	9.62	89.3	Interval method	0.74	1.77	2.84	timeout
SUB	0.11	0.29	2.06	21.0	SUB	0.15	0.31	0.46	0.61

We remark that many more types of synthetic datasets were considered, as points in convex position, points which are vertices of regular n -gons, points on

a grid, etc, see Fig. 5.13. Most of the useful information about the behavior of the algorithms can be extracted by experimenting on UNIF-1, UNIF-2 and TSPLIB, so we chose to mainly experiment and analyze only these. As an example, the running times for Algorithm 4 with PCA for fixed $n = 100$ and $\varepsilon = 10^{-6}$ are given in following table.

$n = 100, \varepsilon = 10^{-6}$	clusters	convex position	n -gon	UNIF-1	UNIF-2
running times	0.37	0.26	0.33	0.34	0.33

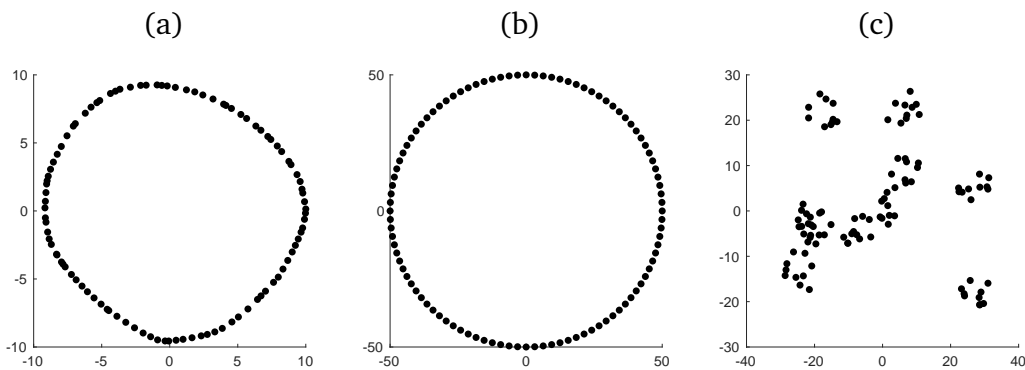


Figure 5.13. (a) 100 points in convex position (b) points of a regular 100-gon (c) 100 points split among 10 clusters

Summary on the Fermat point. We make an overall comparison of Algorithm 3, Algorithm 4 with the PCA, and Algorithm 5, illustrated in Fig. 5.12. The running time of all methods shows a linear dependency on n , but there are big differences regarding the dependency on ε . Overall, Algorithm 5 performs well in all cases, but due to the linear convergence of Weiszfeld's point sequence, it cannot converge faster as ε decreases. In contrast, Algorithm 4 takes more time in the subdivision phase, but once the Newton tests succeeds, the algorithm terminates very quickly. So, it does not exhibit almost any changes in the running time for decreasing ε . This makes it favorable when a high precision approximate solution is required. It is also very fast in UNIF-2 instances and outperforms Algorithm 5. Summarizing, we suggest to use Algorithm 4 in small dimensional spaces and for small ε due to its eventual quadratic convergence in ε . On the other hand, the subdivision methods take exponential time in d , therefore, we suggest to use Algorithm 5 for higher dimensional spaces.

n -ellipses. Finally, we evaluated the runtime of n -ellipses algorithm. In Fig. 5.14(b) we evaluate the dependency on n . In order to keep the length of the curve almost constant we choose the radii $r = \frac{(10\sqrt{2}+2)n}{2}$. The bounding box used is $[-2, 12]^2$. In Fig. 5.14(c) we analyze the dependency on the length of the n -ellipse. The bounding box is fixed and we experimented with different radii such that the lengths of the curve differ by a factor of $3/2$. The runtime shows a linear dependency on n , as expected, and it also shows a linear dependency on the length of curve. This can be justified, as covering an n -ellipse of length l with boxes of width ε takes $O(l/\varepsilon)$ many boxes.

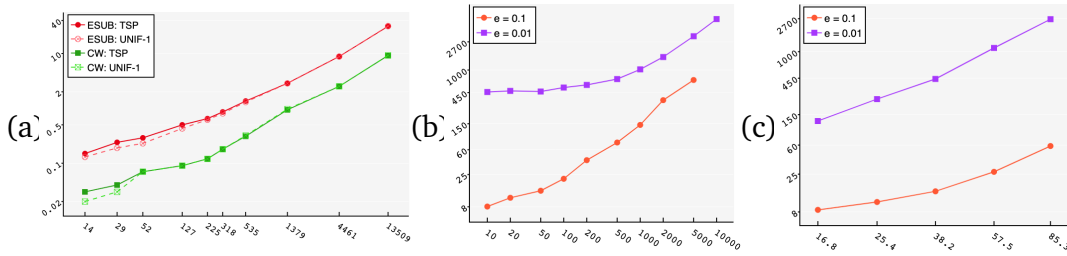


Figure 5.14. (a) A comparison of TSP data sets (filled shapes) with Unif-1 (empty shapes, dashed curve) for both Algorithm 4 (\bullet ESUB) and Algorithm 5 (\blacksquare CW). Fermat point with time as a function of n . (b),(c) n -ellipse on Unif-2 with time as a function of (b) n and (c) the length of the n -ellipse. Two ε approximations with $\varepsilon = 0.1$ (\bullet) and $\varepsilon = 0.01$ (\blacksquare) have been computed.

5.6 Details on box approximations

For this section we need to introduce a bit more notation. Let $()^T$ denote the transpose operation. The Hessian $\nabla^2 f : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ of f is given by the matrix $\nabla^2 f(\mathbf{x}) = (f_{ij}(\mathbf{x}))_{i,j=1}^d$ where $f_{ij} = \partial_i \partial_j f$.

The box approximations described in this section generalize for higher dimensions. For simplicity we describe them in \mathbb{R}^2 .

5.6.1 Box approximation of the gradient $\nabla \varphi$

For any point $p = (p_x, p_y)^T$, let $\sin(p) := p_x/\|p\|$ and $\cos(p) := p_y/\|p\|$. Clearly,

$$\nabla \varphi(p) = \left(\begin{array}{c} \sum_{a \in A} w(a) \sin(p - a) \\ \sum_{a \in A} w(a) \cos(p - a) \end{array} \right).$$

We want to develop formulas for $\sin(B - \mathbf{a})$ and $\cos(B - \mathbf{a})$. By symmetry, we consider only $\sin(B - \mathbf{a})$. The four corners of B are given by $m_B + \frac{\omega(B)}{2} \begin{pmatrix} \pm 1 \\ \pm 1 \end{pmatrix}$. Let $\text{Corners}(B)$ denote this set of four points. Then

$$\sin(B - \mathbf{a}) = \begin{cases} [-1, 1] & \text{if } \mathbf{a} \in B, \\ [\min(\sin(\text{Corners}(B) - \mathbf{a})), 1] & \text{if } \mathbf{a} \text{ is left of } B, \\ [-1, \max(\sin(\text{Corners}(B) - \mathbf{a}))] & \text{if } \mathbf{a} \text{ is right of } B, \\ [\min(\sin(\text{Corners}(B) - \mathbf{a})), \max(\sin(\text{Corners}(B) - \mathbf{a}))] & \text{else.} \end{cases}$$

In other words, $\sin(B - \mathbf{a})$ can be computed from the sinus of at most four angles. Similarly for $\cos(B - \mathbf{a})$.

Now, we extend these formulas: for instance,

$$\square \nabla \varphi(B) = \left(\frac{\sum_{\mathbf{a} \in A} w(\mathbf{a}) \sin(B - \mathbf{a})}{\sum_{\mathbf{a} \in A} w(\mathbf{a}) \cos(B - \mathbf{a})} \right).$$

The following is immediate:

Lemma 5.6.1. $\square \nabla \varphi$ is a soft predicate, i.e. it is conservative and convergent.

Evaluating $\square \nabla \varphi$ as described above gives a very good soft version of $\nabla \varphi$. Evaluating a function for all corners of a box takes exponential time in d . If the number of dimensions is higher, one can instead directly apply interval arithmetic to compute such soft versions in $O(nd)$ time.

5.6.2 Box approximation of φ

We use the concept of a Lipschitz constant in order to derive a box approximation of φ . We call $L(B)$ a Lipschitz constant for box B if $\forall p, q \in B : |\varphi(p) - \varphi(q)| \leq L(B) \cdot \|p - q\|$. A trivial Lipschitz constant is W because it bounds the maximum length of the gradient:

$$\|\nabla \varphi(p)\| \leq \sum_{\mathbf{a} \in A} w(\mathbf{a}) \left\| \begin{pmatrix} \sin(p - \mathbf{a}) \\ \cos(p - \mathbf{a}) \end{pmatrix} \right\| = \sum_{\mathbf{a} \in A} w(\mathbf{a}) = W$$

Definition 5.6.2. We use $\square \varphi(B)$ as a box approximation of $\varphi(B)$ where:

$$\square \varphi(B) = [\varphi(m_B) - L(B) \cdot r_B, \varphi(m_B) + L(B) \cdot r_B]$$

Lemma 5.6.3. $\square \varphi(B)$ is a soft predicate, i.e. it is conservative and convergent.

Proof. The $L(B)$ is a Lipschitz constant of φ on box B , i.e. $\forall p \in B$:

$$|\varphi(p) - \varphi(m_B)| \leq L(B) \cdot r_B$$

This implies $\varphi(p) \in [\varphi(m_B) - L \cdot r_B, \varphi(m_B) + L \cdot r_B]$ and hence $\square\varphi(B)$ is conservative. Let B_i be a sequence of boxes, which converges to a point. This implies $r_{B_i} \rightarrow 0$. The Lipschitz constant L can be bounded from above by W . Thus, $\omega(\square\varphi(B_i)) \leq 2W \cdot r_{B_i} \rightarrow 0$. \square

Using the Lipschitz constant W within all boxes B can result in very bad box approximations. Consider boxes near the Fermat point, for which the gradient of φ at every point is almost 0. In the rest of this section we compute a better Lipschitz constant for each individual box.

We partition the set of foci $A = A_1 \dot{\cup} A_2$ into foci which are "far" or "close" to box B :

$$\forall \mathbf{a} \in A_1 : \left\| \begin{pmatrix} \sin(B - \mathbf{a}) \\ \cos(B - \mathbf{a}) \end{pmatrix} \right\| \in [-1, 1] \quad \text{and} \quad \forall \mathbf{a} \in A_2 : \left\| \begin{pmatrix} \sin(B - \mathbf{a}) \\ \cos(B - \mathbf{a}) \end{pmatrix} \right\| \notin [-1, 1]$$

The length of an interval vector $I = (I_x, I_y)$ is computed by $\|I\| = \sqrt{I_x^2 + I_y^2}$, where we define the square root of an interval $J = [J_1, J_2]$ by:

$$\sqrt{J} = \begin{cases} [0, \sqrt{\max\{|J_1|, |J_2|\}}] & \text{if } 0 \in J \\ [\sqrt{\min\{|J_1|, |J_2|\}}, \sqrt{\max\{|J_1|, |J_2|\}}] & \text{if } 0 \notin J. \end{cases}$$

A box approximation of the length of the gradient of φ can then be achieved by:

$$\square\|\nabla\varphi(B)\| = \left\| \sum_{\mathbf{a} \in A_1} w(\mathbf{a}) \begin{pmatrix} \sin(B - \mathbf{a}) \\ \cos(B - \mathbf{a}) \end{pmatrix} \right\| + \left[-\sum_{\mathbf{a} \in A_2} w(\mathbf{a}), \sum_{\mathbf{a} \in A_2} w(\mathbf{a}) \right]$$

The maximal length of the gradient within box B is a Lipschitz constant of φ within box B . Hence, $L(B) = \max \square\|\nabla\varphi(B)\|$ can be used as Lipschitz constant for box B .

5.6.3 Box approximation of the Hessian $\nabla^2\varphi$

For any $p \in \mathbb{R}^2 \setminus A$ it holds:

$$\nabla^2\varphi(p) = \begin{pmatrix} \sum_{\mathbf{a} \in A} w(\mathbf{a}) \frac{(p_y - a_y)^2}{\|p - \mathbf{a}\|^3} & -\sum_{\mathbf{a} \in A} w(\mathbf{a}) \frac{(p_x - a_x)(p_y - a_y)}{\|p - \mathbf{a}\|^3} \\ -\sum_{\mathbf{a} \in A} w(\mathbf{a}) \frac{(p_x - a_x)(p_y - a_y)}{\|p - \mathbf{a}\|^3} & \sum_{\mathbf{a} \in A} w(\mathbf{a}) \frac{(p_x - a_x)^2}{\|p - \mathbf{a}\|^3} \end{pmatrix}.$$

Definition 5.6.4. We define the box approximation of $\nabla^2\varphi(B)$, denoted $\square\nabla^2\varphi(B)$ as follows.

$$\square\nabla^2\varphi(B) = \begin{pmatrix} \sum_{a \in A} w(a) \frac{(B_y - a_y)^2}{[\|m_B - a\| - r, \|m_B - a\| + r]^3} & - \sum_{a \in A} w(a) \frac{(B_x - a_x)(B_y - a_y)}{[\|m_B - a\| - r, \|m_B - a\| + r]^3} \\ - \sum_{a \in A} w(a) \frac{(B_x - a_x)(B_y - a_y)}{[\|m_B - a\| - r, \|m_B - a\| + r]^3} & \sum_{a \in A} w(a) \frac{(B_x - a_x)^2}{[\|m_B - a\| - r, \|m_B - a\| + r]^3} \end{pmatrix}.$$

The following lemma is again immediate.

Lemma 5.6.5. $\square\nabla^2\varphi$ is conservative and convergent.

5.7 Conclusion

In this work, we mainly focused on finding ε -approximate Fermat points, in a strong sense $\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| \leq \varepsilon$, which had not been considered before. This approximation can also be used to derive an ε -approximation of the Fermat radius. This was done using a simple-to-implement subdivision approach. All of our algorithms are certified in the sense of interval arithmetic. Moreover, we certified the famous point-sequence algorithm of Weiszfeld [1937] to guarantee that it does find an ε -approximate Fermat point. We also designed an algorithm to construct ε -approximate n -ellipses. The simplicity and efficiency of our algorithms were evaluated experimentally for $d = 2$.

There are many directions for further research. One is to derive algorithmic complexity bounds. Our intuition regarding the time complexity of our algorithms was affirmed by the experimental runtime evaluation. Such bounds are rare for iterative numerical algorithms. There has been considerable success in the area of root isolation Burr et al. [2009]; Burr [2016] where the idea of “continuous amortization” should also apply here. Further, we expect the usage of the Hansen-Sengupta Newton operator to result in a speedup.

Regarding the construction of n -ellipses, it would be interesting to design an alternative algorithm based on curve-tracing. This could improve the runtime once a starting point on the n -ellipse is found.

Another direction is related to *Voronoi diagrams*. From one perspective, it is interesting to approximate the Voronoi diagram, where the sites are n -ellipses; so far only 2-ellipses have been studied Emiris et al. [2006]. From a different perspective, if the sites are sets of foci (each associated with a Fermat distance function) it is interesting to compute their Voronoi diagram, defined as the minimization diagram of the Fermat distance functions. This is a *min-sum* diagram in the context of cluster Voronoi diagrams, see e.g., Huttenlocher et al. [1993];

Papadopoulou [2004]. We believe that subdivision methods augmented with root boxes, similar to Bennett et al. [2016], would be applicable to these problems.

Chapter 6

Bichromatic discrepancy

Chapter 6 is based on parts of the following paper:

- Man-Kwun Chiu, Matias Korman, Martin Suderland, and Takeshi Tokuyama. [2022]
Distance bounds for high dimensional consistent digital rays and 2-d partially-consistent digital rays.
Discrete & Computational Geometry pp. 1–43.

How uniform can one distribute red and blue points in the unit square, such that the difference between the number of points in a sample area is close to the expected difference? See an example in Fig. 6.1.

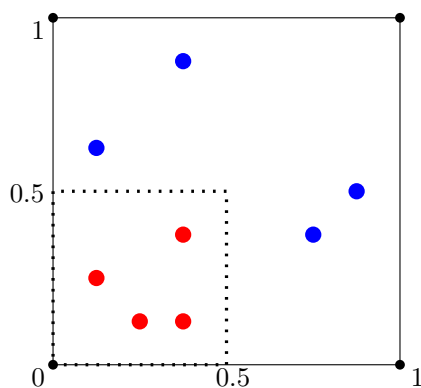


Figure 6.1. Red and blue points sets of cardinality 4 with high discrepancy. The indicated rectangle contains only red points, even though the expected difference between red and blue points in such a rectangle would be 0.

We formally define the bichromatic discrepancy as follows. Let R and B be a set of red and blue points in the unit square, respectively. Let $m := |B| - |R|$

and assume $m \geq 0$. For any set P of points in the unit square and $x, y \in [0, 1]$ let $P[x, y]$ be the number of points in $P \cap [0, x] \times [0, y]$. For any two real numbers $0 \leq x, y \leq 1$ we define the discrepancy of R and B at (x, y) as

$$D_{R,B}(x, y) = mxy - (B[x, y] - R[x, y]). \quad (6.1)$$

The *discrepancy* of R and B is simply defined as

$$D_{R,B}^* = \max_{(x,y) \in [0,1]^2} |D_{R,B}(x, y)|$$

(i.e., the highest discrepancy we can achieve among all possible rectangles). The discrepancy $D_{R,B}^*$ of a two-colored pointset is high if and only if there is an axis-aligned rectangle with the origin as corner in which the difference of the cardinalities is far from the expected difference.

This is a natural extension of the concept of discrepancy. Indeed, the classic definition of (monochromatic) discrepancy is the particular case in which $R = \emptyset$ (see Matoušek [1999] for a detailed survey of this concept and its many applications). To the best of our knowledge, the extension of discrepancy to chromatic settings is largely unexplored. Beck [1981] considered coloring an uncolored pointset so as to minimize the chromatic discrepancy (obtaining $O(\log^4 n)$ upper and $\Omega(\log^4 n)$ lower bounds for the difference between the number of red and blue points within axis-aligned rectangles. If the rectangles are allowed to have arbitrary orientations instead then they derive an $O(n^{1/2+\varepsilon})$ upper and $\Omega(n^{1/4-\varepsilon})$ lower bound for any $\varepsilon > 0$. Dobkin et al. [1996] introduce algorithms to find the maximum discrepancy for a given pointset (under different definitions of bichromatic discrepancy).

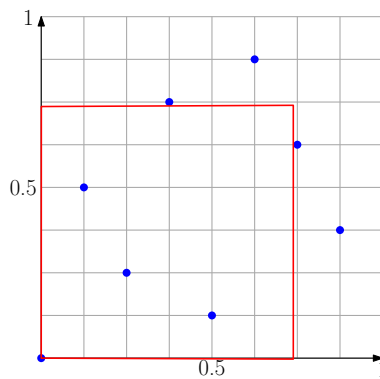


Figure 6.2. The discrepancy of the red rectangle is roughly $|8 \times 0.75^2 - 4| = 0.5$.

6.1 Discrepancy lower bound

In this section we show that the bichromatic discrepancy cannot be arbitrarily small unless the sets of red and blue points are of same cardinality. In that case, trivially choosing $R = B$ results in a setting with 0 discrepancy.

Theorem 6.1.1 (Bichromatic discrepancy). *There exists a constant $c > 0$ such that for any set R and B of points with $|B| \geq |R|$,*

$$D_{R,B}^* \geq c \left(\frac{(|B| - |R|) \cdot \log(|B| + |R|)}{|B| + |R|} \right).$$

Note that the aforementioned bound is tight for the trivial case $|R| = |B|$ but also for $R = \emptyset$, in which we get the classic two dimensional discrepancy result. For the latter case there are several proofs for the $\Omega(\log |B|)$ lower bound and its tightness, see Matoušek [1999] for a detailed survey. In order to extend the bound for the case of $R \neq \emptyset$, we make minor changes to Schmidt's proof Schmidt [1972]. We start by using an auxiliary function G (defined below) and combining it with the trivial inequality

$$\begin{aligned} & \int_{(x,y) \in [0,1]^2} D_{R,B}(x,y) G(x,y) dx dy \\ & \leq \max_{(x,y) \in [0,1]^2} |D_{R,B}(x,y)| \int_{(x,y) \in [0,1]^2} |G(x,y)| dx dy \end{aligned}$$

to obtain

$$D_{R,B}^* = \max_{(x,y) \in [0,1]^2} |D_{R,B}(x,y)| \geq \frac{\int D_{R,B} G}{\int |G|}.$$

Note that for simplicity in the notation we removed the integration limits as we always integrate over all points (x, y) within the unit square. Our definition of G is identical to the one used by Schmidt: Let $m = \lceil \log_2(b+r) \rceil + 1$ and observe that, by definition of m we have $2(b+r) \leq 2^m \leq 4(b+r)$. Only in this section we use the variable name “ m ” in a different sense in order to keep the notation similar to Matoušek [1999]. For any $j \in \{0, \dots, m\}$ we define function $f_j : [0, 1]^2 \rightarrow \{-1, 0, 1\}$ as follows: subdivide the unit square with 2^j equally spaced vertical lines and 2^{m-j} horizontal lines.

For any value of j we subdivide the unit square into rectangles of area 2^{-m} (larger values of j will result in thinner but wider rectangles). Let A be a rectangle

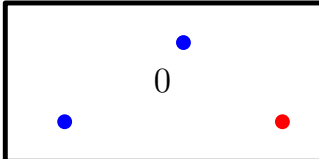
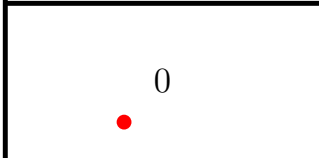
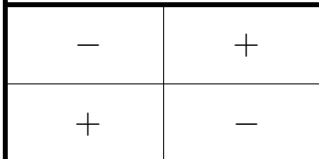
		-	+
		+	-
		-	+
		+	-
-	+		
+	-		
-	+	-	+
+	-	+	-

Figure 6.3. Illustration of f_j for $m = 3$ and $j = 1$.

of subdivision associated to f_j . We define f_j within the rectangle to be 0 if A contains any point of $R \cup B$. If A does not have neither red nor blue points, we further subdivide it into four congruent quadrants. The function value of f_j is equal to 1 in the upper right and lower left quadrants, and -1 in upper left and lower right quadrants, see Fig. 6.3.

Then, we define G as $G = (1 + cf_0)(1 + cf_1) \dots (1 + cf_m) - 1$, where $c \in (0, 1)$ is a small constant (whose value will be chosen afterwards). Note that G can also be expressed as $G = G_1 + \dots + G_m$, where

$$G_k = c^k \sum_{0 \leq j_1 \leq \dots \leq j_m \leq m} f_{j_1} f_{j_2} \dots f_{j_k}.$$

Schmidt showed that $\int |G| \leq 2$ (regardless of the value of m). Thus, we now focus in giving a lower bound for $\int D_{R,B} G$.

Lemma 6.1.2. *There exists a constant c_1 such that $\int D_{R,B} G_1 \geq cc_1 \frac{b-r}{b+r} \log(b+r)$.*

Proof. By definition of G_1 we have $\int D_{R,B} G_1 = c \sum_{j=0}^m \int D_{R,B} f_j$. Thus, it suffices to show that for any value of j it holds that $\int D_{R,B} f_j \geq c' \frac{b-r}{b+r}$ (for some other constant $c' > 0$).

Recall that, when defining f_j , we subdivided the unit square into at least $2(b+r)$ rectangles. For the rectangles that contain at least one point of $R \cup B$, f_j

is set to zero, and thus they do not contribute to the integral. Since we have $b + r$ many points, we know that there must exist at least $b + r$ rectangles that do not contain any point of R or B . Let A be any such rectangle, and let $A_{SW}, A_{NW}, A_{SE}, A_{NE}$ be the four subquadrants of A (where the subindex refers to the cardinal position of the quadrant). Recall that f_j is equal to 1 for any point of $A_{SW} \cup A_{NE}$ and -1 for points of $A_{SE} \cup A_{NW}$.

Let w and h be vectors defined by the horizontal and vertical sides of A_{SW} , respectively. Observe that their lengths are 2^{-j-1} and 2^{j-m-1} , respectively. Then, we have

$$\begin{aligned} & \int_A f_j D_{R,B} \\ &= \int_{A_{SW}} D_{R,B} - \int_{A_{NW}} D_{R,B} + \int_{A_{NE}} D_{R,B} - \int_{A_{SE}} D_{R,B} \\ &= \int_{A_{SW}} [D_{R,B}(x, y) + D_{R,B}(x + w, y + h) - D_{R,B}(x, y + h) \\ & \quad - D_{R,B}(x + w, y)] dx dy. \end{aligned}$$

If we apply the definition of $D_{R,B}$ (Equation (6.1)) to the four terms inside the integral we get

$$\begin{aligned} \int_A f_j D_{R,B} &= \int_{A_{SW}} ((b - r)[xy + (x + w)(y + h) - x(y + h) - (x + w)y]) dx dy \\ & \quad - \int_{A_{SW}} (B[x, y] + B[x + w, y + h] - B[x, y + h] - B[x + w, y]) dx dy \\ & \quad + \int_{A_{SW}} (R[x, y] + R[x + w, y + h] - R[x, y + h] - R[x + w, y]) dx dy. \end{aligned}$$

Observe that we are integrating twice positively and twice negatively over almost identical functions. In fact, the terms of the first integral all cancel out except along the rectangle $[x, x + w) \times [y, y + h)$. Similarly, when we look at the second and third terms, the contribution of any point in $R \cup B$ is cancelled out unless it is in the rectangle $[x, x + w) \times [y, y + h)$. However, by definition of A there are no such points. Thus, we obtain

$$\int_A f_j D_{R,B} = \int_{A_{SW}} (b - r)w \cdot h dx dy = \int_{A_{SW}} (b - r)2^{-m-2} dx dy = (b - r)2^{-2m-4}$$

That is, when we integrate $f_j D_{R,B}$ over a rectangle A containing no point of $R \cup B$, the result is $(b-r)2^{-2m-4}$. We know that there are at least $b+r$ rectangles not containing points of $R \cup B$, thus their contribution is at least $\frac{(b+r)(b-r)}{2^{2m+4}} = \frac{(b+r)(b-r)}{2^m 16 \cdot 2^m} \geq \frac{1}{4} \frac{(b-r)}{16 \cdot 4(b+r)} = \Omega\left(\frac{b-r}{b+r}\right)$. \square

Lemma 6.1.3. *There exists a constant c_2 such that*

$$\sum_{k=2}^m \int D_{R,B} G_k \leq c^2 c_2 \frac{b-r}{b+r} \log(b+r)$$

Proof. Recall that $G_k = c^k \sum_{0 \leq j_1 < j_2 < \dots < j_k \leq m} f_{j_1} \dots f_{j_k}$. Fix any valid set of indices and consider the value of $\int f_{j_1} \dots f_{j_k} D_{R,B}$.

As shown in Matoušek [1999], function $f_{j_1} \dots f_{j_k}$ is largely defined by f_{j_1} and f_{j_k} . Indeed, if we overlay the rectangular partition defined by functions f_{j_1}, \dots, f_{j_k} we obtain a grid of rectangles whose width is 2^{-j_k} and height $2^{-(m-j_1)}$. In each of these rectangles, the function is zero (if any of the rectangles associated to the f_{j_i} functions contains a point of $R \cup B$), or is further subdivided into four equal sized quadrants and in each one it is $+1$ or -1 alternatively.

Let A be one of the rectangles of the refined grid. As shown in Lemma 6.1.2, we have that

$$\int_A f_{j_1} \dots f_{j_k} D_{R,B} = \tau (b-r) 2^{-2(m+j_k-j_1)-4},$$

where $\tau \in \{-1, 1\}$. This extra term appears because the product of the different functions involved can change the sign of each of the four quadrants. In any case, we have $\int_A f_{j_1} \dots f_{j_k} D_{R,B} \leq (b-r) 2^{-(m+g)-4}$ where $g = j_k - j_1$.

By the way the grid is constructed, there are $2^{m-j_1} \times 2^{j_k} = 2^{m+g}$ many rectangles, and thus we conclude that $\int f_{j_1} \dots f_{j_k} D_{R,B} \leq (b-r) 2^{-m-g-4}$. In order to obtain a bound $\int D_{R,B} G_k$ we sum over all possible indices.

$$\begin{aligned} \int D_{R,B} G_k &= c^k \sum_{0 \leq j_1 < j_2 < \dots < j_k \leq m} \int f_{j_1} \dots f_{j_k} D_{R,B} \\ &\leq \frac{c^k (b-r)}{2^{m+4}} \sum_{0 \leq j_1 < j_2 < \dots < j_k \leq m} 2^{-(j_k-j_1)}. \end{aligned}$$

Note that in the sum, the indices j_2, \dots, j_{k-1} do not matter. Thus, we group the terms by the gap between the indices j_1 and j_k (say, if $j_1 = 3$ and $j_k = 7$ the gap is 4). Note that the minimum gap is at least $k-1$ (since otherwise we do not have

enough space to choose the $k - 2$ indices in between) and at most m . Once we have a gap of g there are $m - g$ options for index j_1 .

$$\begin{aligned} \int D_{R,B} G_k &\leq \frac{c^k(b-r)}{2^{m+4}} \sum_{g=k-1}^m \sum_{j_1=0}^{m-g} \sum_{j_1 < j_2 < \dots < j_{k-1} < j_1 + g} 2^{-g} \\ &= \frac{c^k(b-r)}{2^{m+4}} \sum_{g=k-1}^m \sum_{j_1=0}^{m-g} \binom{g-1}{k-2} 2^{-g} \\ &\leq \frac{c^k(b-r)m}{2^{m+4}} \sum_{g=k-1}^m \binom{g-1}{k-2} 2^{-g}. \end{aligned}$$

In order to bound the sum over all G_k from above, we first reorder the summation order.

$$\begin{aligned} \sum_{k=2}^m \int D_{R,B} G_k &\leq \sum_{k=2}^m \frac{c^k(b-r)m}{2^{m+4}} \sum_{g=k-1}^m \binom{g-1}{k-2} 2^{-g} \\ &= \frac{(b-r)m}{2^{m+4}} \sum_{g=1}^m 2^{-g} \sum_{k=2}^{g+1} c^k \binom{g-1}{k-2} \\ &= \frac{(b-r)m}{2^{m+4}} \sum_{g=1}^m 2^{-g} c^2 \sum_{i=0}^{g-1} \binom{g-1}{i} c^i \\ &= \frac{(b-r)m}{2^{m+4}} \sum_{g=1}^m 2^{-g} c^2 (1+c)^{g-1} \\ &= \frac{(b-r)mc^2}{2^{m+5}} \sum_{g=1}^m \left(\frac{1+c}{2}\right)^{g-1}. \end{aligned}$$

The sum contains the first terms of the geometric sum $\sum_{g=1}^{\infty} \left(\frac{1+c}{2}\right)^{g-1} \leq \frac{2}{1-c}$ (for any $c < 1$). In particular, if we set $c \leq 1/2$ we can bound the partial sum by 4 from above. Recall that $m = \Theta(\log(b+r))$ and $2^m = \Theta(b+r)$. Thus, the lemma is proven. \square

Corollary 6.1.4. *There exists a constant $\kappa > 0$ such that*

$$\int D_{R,B} G \geq \kappa \left(\frac{(b-r) \cdot \log(b+r)}{b+r} \right)$$

Proof. Apply the inequality $\int (A + B) \geq \int A - \int |B|$ and Lemmas 6.1.2 and 6.1.3 to obtain:

$$\int D_{R,B}G = \int D_{R,B}G_1 + \sum_{k=2}^m \int D_{R,B}G_k \geq c(c_1 - cc_2) \left(\frac{(b-r) \cdot \log(b+r)}{b+r} \right)$$

Note that Lemmas 6.1.2 and 6.1.3 holds for any value of c such that $c \in (0, 1/2]$. By choosing a sufficiently small value of c (say, $c = \min\{\frac{1}{2}, \frac{c_1}{2c_2}\}$) we obtain

$$\int D_{R,B}G \geq \frac{cc_1}{2} \left(\frac{(b-r) \cdot \log(b+r)}{b+r} \right)$$

□

6.2 Point sets with constant discrepancy

In this section we construct red R and blue B point sets such that the absolute value of their discrepancy is 1. Let $m > 0$ be the difference between the number of blue and red points as defined in Section 6.1. Our construction has $\Theta(m^2)$ many points. Afterwards we also prove that a discrepancy of 1 cannot be achieved with $o(m^2)$ many points.

We first describe a specific configuration of points, called staircase.

Definition 6.2.1. *A staircase is a sequence of alternating blue and red points (p_1, p_2, \dots, p_n) in the unit square. It starts and ends with a blue point. Moreover for every red point p_i , the blue point p_{i-1} has smaller x -coordinate and the same y -coordinate. The blue point p_{i+1} has the same x -coordinate and smaller y -coordinate.*

Given a staircase, we can define a curve by connecting consecutive points on the staircase. Additionally we add a vertical segment at the beginning and a horizontal segment at the end, in order to connect the curve to the boundary of the unit square, see Figure 6.4. We will also use the term “staircase” for this curve.

Assume that a set of blue and red points forms one staircase. Then the curve induced by the staircase splits the unit square into two parts. The set of points (x, y) to the bottom-left of the staircase satisfies $B[x, y] - R[x, y] = 0$ whereas the set of points to the top-right of the staircase satisfies $B[x, y] - R[x, y] = 1$,

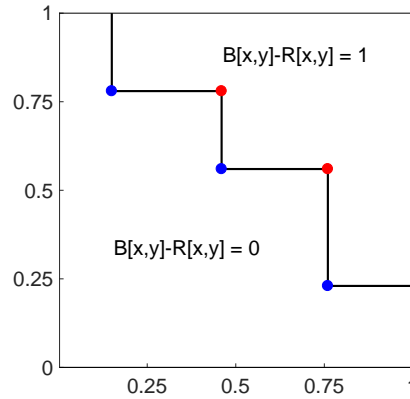


Figure 6.4. A Staircase

see Figure 6.4. When the set of blue and red points can be decomposed into many staircases, then we can easily compute the value $B[x, y] - R[x, y]$ by counting how many staircases are to the bottom-left of point (x, y) .

Recall the definition of discrepancy of R and B at point (x, y) :

$$D_{R,B}(x, y) = mxy - (B[x, y] - R[x, y]).$$

The first term mxy represents the expected difference between the numbers of blue and red points in the axis-aligned rectangle with corner points $(0, 0)$ and (x, y) . Every point (x, y) along the curves $C_i := \{(x, y) \in [0, 1]^2 \mid x \cdot y = \frac{i}{m}\}$, where $i \in \{0, 1, \dots, m\}$, describes a rectangle $[0, x] \times [0, y]$ in which we expect i many blue points more than red points. Figure 6.5 illustrates the curves C_i in black for $m = 7$.

The idea of our construction is to approximate the level curves $C_{i-0.5}$ by staircases, where $i \in \{1, \dots, m\}$. We will construct m staircases such that the staircase approximating $C_{i-0.5}$ is between C_{i-1} and C_i . This guarantees that the discrepancy $D_{R,B}^*$ is at most 1.

We describe how we construct the staircase which approximates $C_{i-0.5}$. We start with a blue point at the intersection of the two curves C_{i-1} and $x = y$. This is the blue point $(\frac{\sqrt{i-1}}{\sqrt{m}}, \frac{\sqrt{i-1}}{\sqrt{m}})$. Starting from there we move horizontally to the right until we hit the curve C_i at the point $(\frac{i}{\sqrt{i-1}\sqrt{m}}, \frac{\sqrt{i-1}}{\sqrt{m}})$. We add a red point here. Then we move vertically down until we hit C_{i-1} and put a blue point. We continue in this fashion, i.e. from a blue point on C_{i-1} we move horizontally to the right and put a red point on C_i . From a red point on C_i we move vertically down and put a blue point on C_{i-1} . The blue points will have the coordinates $(\frac{i^k}{(i-1)^{k-0.5}\sqrt{m}}, \frac{(i-1)^{k+0.5}}{i^k\sqrt{m}})$ and

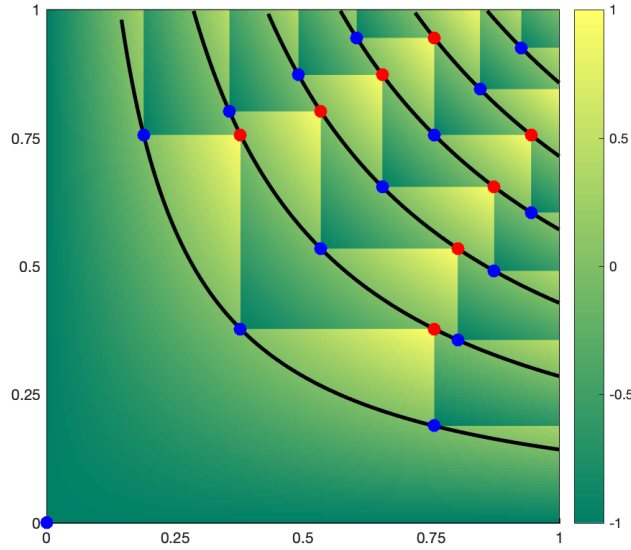


Figure 6.5. Staircase approximation for $m = 7$.

The curves C_i are drawn in black. The brightness of the green color encodes the value of the discrepancy $D_{R,B}(x, y)$ at each point (x, y) . The discrepancy values range between -1 and 1 as shown on the right hand side. The staircases can be seen at the discontinuity of the discrepancy function. At each staircase the discrepancy function changes its value by 1.

the red points have the coordinates $(\frac{i^{k+1}}{(i-1)^{k+0.5}\sqrt{m}}, \frac{(i-1)^{k+0.5}}{i^k\sqrt{m}})$, where $k \in \{0, 1, 2, \dots\}$. We stop this construction when we leave the unit square, i.e., we look for the largest k such that the blue point $(\frac{i^k}{(i-1)^{k-0.5}\sqrt{m}}, \frac{(i-1)^{k+0.5}}{i^k\sqrt{m}})$ is still contained in $[0, 1]^2$. The maximum value for k is

$$k = \left\lfloor \frac{\log\left(\frac{\sqrt{m}}{\sqrt{i-1}}\right)}{\log\left(\frac{i}{i-1}\right)} \right\rfloor$$

for $i \geq 2$ and $k = 0$ for $i = 1$. So far we described how we construct the staircases on the side $y \leq x$. We add red and blue points on the side $y > x$ to make the construction symmetric to the line $y = x$. Figure 6.5 illustrates our construction, which we call the symmetric greedy staircase construction.

Observation 6.2.2. *The points of the symmetric greedy staircase construction with*

m stairs are

$$B = \left\{ \left(\frac{i^k}{(i-1)^{k-0.5} \cdot \sqrt{m}}, \frac{(i-1)^{k+0.5}}{i^k \cdot \sqrt{m}} \right) \mid i \in [1, m] \text{ and } -k^*(i) \leq k \leq k^*(i) \right\}$$

$$R = \left\{ \left(\frac{i^{k+1}}{(i-1)^{k+0.5} \cdot \sqrt{m}}, \frac{(i-1)^{k+0.5}}{i^k \cdot \sqrt{m}} \right) \mid i \in [1, m] \text{ and } -k^*(i) \leq k \leq k^*(i) - 1 \right\}$$

where $[1, m] = \{1, 2, \dots, m\}$ and

$$k^*(i) = \begin{cases} 0 & \text{if } i = 1 \\ \left\lfloor \frac{\log\left(\frac{\sqrt{m}}{\sqrt{i-1}}\right)}{\log\left(\frac{i}{i-1}\right)} \right\rfloor & \text{if } i \neq 1. \end{cases}$$

There are $2k^*(i) + 1$ (resp. $2k^*(i)$) many blue (resp. red) points in the i -th staircase.

Theorem 6.2.3. *The symmetric greedy staircase construction has discrepancy 1.*

Proof. Consider any point (x, y) between the i -th and $(i+1)$ -th staircase. It holds that $B[x, y] - R[x, y] = i$. Moreover both staircases are bounded from below by the level curve C_{i-1} and from above by C_{i+1} , which means that $\frac{i-1}{m} \leq x \cdot y \leq \frac{i+1}{m}$. Summarizing, we can bound the discrepancy

$$-1 \leq \underbrace{mxy - (B[x, y] - R[x, y])}_{=D_{R,B}(x,y)} \leq 1.$$

□

Theorem 6.2.4. *The symmetric greedy staircase construction with m staircases has $O(m^2)$ many points.*

Proof. The number of blue points, which are used in our construction, is

$$|B| = \sum_{i=1}^m 1 + 2 \cdot k^*(i) = m + \sum_{i=2}^m 2 \cdot \left\lfloor \frac{\log\left(\frac{\sqrt{m}}{\sqrt{i-1}}\right)}{\log\left(\frac{i}{i-1}\right)} \right\rfloor \leq O(m) + \sum_{i=2}^{m-1} \frac{\log\left(\frac{m}{i}\right)}{\log\left(\frac{i+1}{i}\right)}.$$

We now bound the denominator from below by

$$\log\left(\frac{i+1}{i}\right) = \log\left(1 + \frac{1}{i}\right) \geq \frac{1}{i} - \frac{1}{i^2} = \frac{i-1}{i^2}.$$

Putting the inequalities together, we get:

$$|B| \leq O(m) + \sum_{i=2}^{m-1} \frac{i^2}{i-1} \log\left(\frac{m}{i}\right) \leq O(m) + 2 \sum_{i=2}^{m-1} i \log\left(\frac{m}{i}\right)$$

The continuous function $f(i) = i \log\left(\frac{m}{i}\right)$ has exactly one maximum in the interval $[2, m]$ with a value bounded by $m \log m$ and is monotone on both sides of it. Therefore we can replace the sum by an integral.

$$\begin{aligned} |B| &\leq O(m \log m) + 2 \int_2^{m-1} i \log\left(\frac{m}{i}\right) di \\ &= O(m \log m) + 2 \left[\frac{i^2}{4} \cdot \left(1 + 2 \log\left(\frac{m}{i}\right)\right) \right]_{i=2}^{i=m-1} \\ &= O(m^2). \end{aligned}$$

□

We now show that our construction is tight.

Theorem 6.2.5. *Let B and R be point sets, which can be decomposed into m non-intersecting staircases, and have a discrepancy bounded by a constant ξ . Then $|B| = \Omega(m^2)$.*

Proof. The i -th staircase is bounded from below by the level curve $C_{i-\xi}$ and from above by $C_{i+\xi-1}$ because of the discrepancy constraint. We count how many points are necessary to create the i -th stair. The minimum number can be realized by constructing a stair in a greedy manner between $C_{i-\xi}$ and $C_{i+\xi-1}$ because both curves are convex.

$$\begin{aligned} B &= \left\{ \left(\frac{(i+\xi-1)^k}{m(i-\xi)^{k-1}}, \frac{(i-\xi)^k}{(i+\xi-1)^k} \right) \mid i \in \{1, 2, \dots, m\} \text{ and } 1 \leq k \leq k^*(i) \right\} \\ R &= \left\{ \left(\frac{(i+\xi-1)^{k+1}}{m(i-\xi)^k}, \frac{(i-\xi)^k}{(i+\xi-1)^k} \right) \mid i \in \{1, 2, \dots, m\} \text{ and } 1 \leq k \leq k^*(i) - 1 \right\} \end{aligned}$$

where

$$k^*(i) = \begin{cases} 0 & \text{if } i \leq \xi \\ \left\lfloor \frac{\log\left(\frac{m}{i-\xi}\right)}{\log\left(\frac{i+\xi-1}{i-\xi}\right)} \right\rfloor & \text{if } i > \xi. \end{cases}$$

The number of blue points can therefore be bounded by

$$|B| \geq \sum_{i=1}^m k^*(i) \geq \xi + \sum_{i=\xi+1}^m \left\lfloor \frac{\log\left(\frac{m}{i-\xi}\right)}{\log\left(\frac{i+\xi-1}{i-\xi}\right)} \right\rfloor \geq -O(m) + \sum_{i=1}^{m-\xi} \frac{\log\left(\frac{m}{i}\right)}{\log\left(\frac{i+2\xi-1}{i}\right)}$$

Using the inequality $\log\left(\frac{i+2\xi-1}{i}\right) = \log\left(1 + \frac{2\xi-1}{i}\right) \leq \frac{2\xi-1}{i}$ and comparing the sum with an integral, as done in the proof of Theorem 6.2.4

$$\frac{1}{2\xi-1} \sum_{i=1}^{m-\xi} i \log\left(\frac{m}{i}\right) \geq \frac{1}{2\xi-1} \left(-O(m \log m) + \int_1^{m-\xi} i \log\left(\frac{m}{i}\right) di \right)$$

we can conclude $|B| \geq \Omega(m^2)$. \square

Theorem 6.2.6. *Let B and R be two sets of points whose discrepancy satisfies $D_{R,B}^* < 1$. Then $|B| = \Omega(m^2)$, where $m = |B| - |R|$.*

Proof. Consider the sets $S_i := \{(x, y) \in [0, 1]^2 \mid B[x, y] - R[x, y] = i\}$ for $i \in \{0, 1, \dots, m\}$. Because the discrepancy of the point set B and R is less than 1 we can conclude that

1. the curves C_i are contained in S_i and
2. the points between C_i and C_{i+1} are either contained in S_i or S_{i+1} .

Therefore there exists a curve between C_i and C_{i+1} which is only neighboring S_i to its bottom-left and S_{i+1} to its top-right for each $i \in \{0, 1, \dots, m-1\}$. This curve is a staircase. Hence there exists a staircase between C_i and C_{i+1} for each $i \in \{0, 1, \dots, m-1\}$. Those staircases are non-intersecting because $D_{R,B}^* < 1$. Therefore they consist of at least $\Omega(m^2)$ many points, as shown in Theorem 6.2.5. \square

Theorem 6.2.6 is open if the upper bound $D_{R,B}^* < 1$ is replaced by $O(1)$.

6.3 Application to digitalizing line segments

The lower bound for the bichromatic discrepancy can be applied to the topic of digitalizing line segments. The aim is to construct a digital path $dig(p, q)$ for any two points $p, q \in \mathbb{Z}^d$. Ideally, dig is defined for any pair of points in \mathbb{Z}^d and resembles properties of Euclidean segments, but here we only consider digital paths starting from the origin, i.e. $p = \mathbb{O}$. For brevity we write $dig(p) := dig(\mathbb{O}, p)$.

Definition 6.3.1. *For any $S \subseteq \mathbb{Z}^d$, let $DS(S)$ be a set of digital rays such that $dig(p) \in DS(S)$ for all $p \in S$. We say that $DS(S)$ forms a set of consistent digital rays on S (CDR for short) if for every $p \in S$ it satisfies the following axioms:*

- (S1) *Grid path property:* $dig(p)$ is a path between \mathbb{O} and p under the $2d$ -neighbor topology¹.
- (S2) *Subsegment property:* for any $q \in dig(p)$, $dig(q) \in DS(S)$ and $dig(q) \subseteq dig(p)$.
- (S3) *Prolongation property:* $\exists q \in \mathbb{Z}^d$ such that $dig(q) \in DS(S)$ and $dig(p) \subset dig(q)$.
- (S4) *Monotonicity property:* for all $i \leq d$ and every point $q \in dig(p)$, it holds $0 \leq q_i \leq p_i$ or $p_i \leq q_i \leq 0$.

These axioms give nice properties of digital segments analogous to Euclidean line segments. For example, (S1) and (S2) imply that the intersection of two digital segments is another segment. (S4) implies that the intersection of a segment with an axis-aligned halfspace is a segment (and connected by (S1)), and so on. Note that the five axioms would imply that a CDR is a tree connecting a fixed point o to any other point of \mathbb{Z}^d (see Figure 6.6).

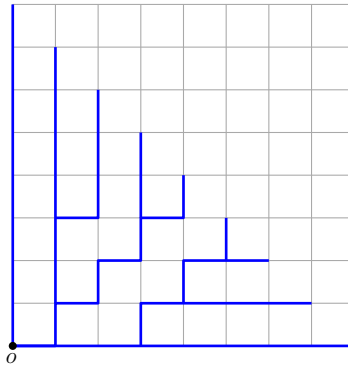


Figure 6.6. An example of a CDR in one quadrant.

Another property that we want from a CDR is that they visually resemble the Euclidean segments. The resemblance between the digital segment $dig(p)$ and the Euclidean counterpart $\overline{\mathbb{O}p}$ is measured using the *Hausdorff* distance. The Hausdorff distance $H(A, B)$ of two objects A and B is defined by $H(A, B) = \max\{h(A, B), h(B, A)\}$, where $h(A, B) = \max_{a \in A} \min_{b \in B} \delta(a, b)$, and $\delta(a, b)$ is the $\|\cdot\|_\infty$ L -infinity norm.

¹The $2d$ -neighbor topology is the natural one that connects to your predecessor and successor in each dimension. Formally speaking, two points are connected if and only if their $\|\cdot\|_1$ distance is exactly one.

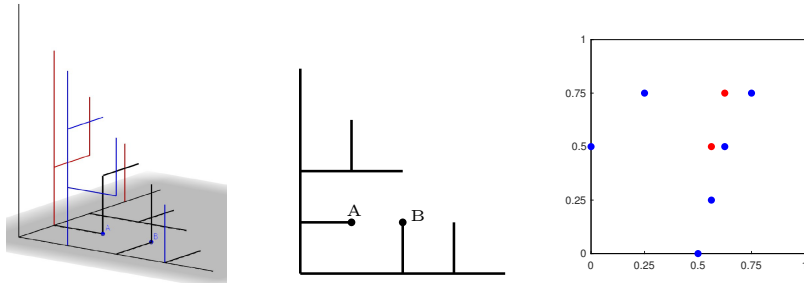


Figure 6.7. (left) A drawing of a CDR in $\mathcal{G}_N^+ \subset \mathbb{Z}^3$ for $N = 4$. Notice that the CDR is a tree whose leaves are at the plane $x + y + z = N$. (middle) A cross section on the xy -plane of the same CDR. Observe that vertices A and B do not extend within the xy -plane. Thus, the subspace is a weak CDR (rather than a proper CDR). (right) A map of the weak CDR into a two-colored point set. Regions with many blue points and few red correspond to portions of the CDR with high error.

The resemblance of a CDR on S is defined as $\max_{(p) \in S} H(\text{dig}(p), \overline{\text{Op}})$ (that is, the largest error created between a digital segment and its Euclidean counterpart). This value is simply referred to as the *error* of the CDR construction. We are interested to see how the error grows as we enlarge our focus of interest. Thus, we limit the domain to the case in which p lies in the L_1 ball of radius N centered at the origin (i.e., $\mathcal{G}_N = \mathbb{Z}^d \cap B_1(o, N)$). Rather than looking for the exact function, we are interested in the asymptotic behavior of the error as a function of N . For simplicity, we will actually restrict ourselves to the positive orthant $\mathcal{G}_N^+ = \mathcal{G}_N \cap H_i$, where $H_i = \{p \in \mathbb{Z}^d : p_i \geq 0\}$ and p_i is the i -th coordinate of p . A construction of a CDR in one orthant can be easily extended to the other orthants.

Note, a CDR in \mathbb{Z}^d , with $d \geq 3$ restricted to the $x - y$ -plane does not need to be a CDR in 2 dimensions. The main reason why a subspace is not a CDR is because of the prolongation property (S3): we require that every segment be extendable, but have no constraints on the dimension in which it does so. In particular, a subspace of a high dimensional CDR need not be a CDR (see an example in Figure 6.7). Subspaces of CDRs are what we call *weak CDR*: it is a construction that almost always behaves like a CDR but some vertices may not satisfy the prolongation property (S3). Each vertex that does not extend is called an *inner leaf*. We will see that the number of inner leaves of a weak CDRs in two dimensions has implications on the Hausdorff error of a (proper) CDRs in higher dimensions.

There has been considerable interest into CDRs and weaker or stronger ver-

sions of it in the recent years, see Luby [1987]; Chun et al. [2009]; Christ et al. [2012]; Chowdhury and Gibson [2015, 2016]; Chiu and Korman [2018]; Chiu et al. [2022]; Gibson-Lopez and Zamarripa [2022].

Combining a transformation from a weak CDR to a two-colored point set, see Fig. 6.7, with the lower bound on the bichromatic discrepancy, Chiu et al. [2022] prove that a small number of inner leaves in a weak CDR induces a high Hausdorff error:

Theorem 6.3.2. *For any $N \in \mathbb{N}$, any weak CDR defined on $\mathcal{G}_N^+ \subset \mathbb{Z}^2$ with κ_2 inner leaves between lines $x_1 + x_2 = \lceil N/2 \rceil$ and $x_1 + x_2 = N$ has $\Omega\left(\frac{N \log N}{N + \kappa_2}\right)$ error.*

On the other hand, they also show that a high number of inner leaves of a CDR in the $x - y$ -plane causes a big Hausdorff error within the other dimensions.

Lemma 6.3.3. *Any CDR defined on $\mathcal{G}_N^+ \subset \mathbb{Z}^d$ with κ_2 inner leaves in $x_1 x_2$ -plane between lines $x_1 + x_2 = \lceil N/2 \rceil$ and $x_1 + x_2 = N$ has $\Omega\left((\kappa_2/N)^{\frac{1}{d-2}}\right)$ error.*

Balancing these two error sources leads to a lower bound of $\Omega(\log^{1/(d-1)} N)$ on the Hausdorff error for any CDR construction in d dimensions:

Theorem 6.3.4. *Any CDR in \mathbb{Z}^d has $\Omega(\log^{1/(d-1)} N)$ error.*

6.4 Conclusion

In Section 6.2 we have constructed red and blue point sets R, B with their discrepancy bounded from above by a constant. If $|B| - |R| = m$, then the construction used points sets of cardinality $\Theta(m^2)$. On the other hand the discrepancy lower bound Theorem 6.1.1

$$D_{R,B}^* \geq c \left(\frac{(|B| - |R|) \cdot \log(|B| + |R|)}{|B| + |R|} \right)$$

ensures that at least $\Omega(m \log m)$ many points are needed. It remains for the future to figure out which (if any) of the two bounds is tight.

We derived the results on the bichromatic discrepancy with the direct application to CDRs in mind. We still believe that the bichromatic discrepancy results are as well of independent interest. For instance, low discrepancy sequences (in the monochromatic case) are often used in the quasi-Monte-Carlo method for numerical integration in a unit cube as described next, see Matoušek [1999]. Let $f : [0, 1]^d \rightarrow \mathbb{R}$ for which the integral $\int_{[0,1]^d} f(x) dx$ needs to be approximated. This can be done by choosing a set of n points $P \subset [0, 1]^d$ and returning the sum

$\frac{1}{n} \sum_{p \in P} f(p)$. If the function f is continuously differentiable then Koksma–Hlawka inequality bounds the difference between the two terms by

$$\left| \int_{[0,1]^d} f(x) dx - \frac{1}{n} \sum_{p \in P} f(p) \right| \leq \frac{1}{n} D_{\emptyset, P}^* \cdot V(f),$$

where $V(f)$ is a generalization for higher dimensions of the variation of one dimensional functions $V(f) = \int_0^1 |f'(x)| dx$. This inequality guarantees faster convergence if the point set P is chosen according to some low discrepancy sequence. It remains to be seen if the new concept of bichromatic discrepancy can be also useful for example for quasi-Monte-Carlo methods.

Bibliography

- Abu-Affash, A. K. and Katz, M. J. [2009]. Improved bounds on the average distance to the Fermat-Weber center of a convex object, *Information Processing Letters* **109**(6): 329–333.
- Agarwal, P. K., de Berg, M., Matousek, J. and Schwarzkopf, O. [1998]. Constructing levels in arrangements and higher order Voronoi diagrams, *SIAM J. Comput.* **27**(3): 654–667.
- Agarwal, P. K. and Sharir, M. [2000]. Arrangements and their applications, *Handbook of computational geometry*, Elsevier, pp. 49–119.
- Aronov, B. [2002]. A lower bound on Voronoi diagram complexity, *Inf. Process. Lett.* **83**(4): 183–185.
- Aronov, B. [2019]. Personal communication.
- Aurenhammer, F. [1987]. Power diagrams: Properties, algorithms and applications, *SIAM Journal on Computing* **16**(1): 78–96.
- Aurenhammer, F. [1990]. A new duality result concerning voronoi diagrams, *Discret. Comput. Geom.* **5**: 243–254.
- Aurenhammer, F. [1991]. Voronoi diagrams - A survey of a fundamental geometric data structure, *ACM Comput. Surv.* **23**(3): 345–405.
- Aurenhammer, F., Drysdale, R. L. S. and Krasser, H. [2006]. Farthest line segment Voronoi diagrams, *Information Processing Letters* **100**(6): 220–225.
- Aurenhammer, F., Jüttler, B. and Paulini, G. [2017]. Voronoi diagrams for parallel halflines and line segments in space, in Y. Okamoto and T. Tokuyama (eds), *28th International Symposium on Algorithms and Computation, ISAAC 2017, December 9-12, 2017, Phuket, Thailand*, Vol. 92 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 7:1–7:10.

- Aurenhammer, F., Klein, R. and Lee, D.-T. [2013]. *Voronoi diagrams and Delaunay triangulations*, World Scientific Publishing Company.
- Badoiu, M., Har-Peled, S. and Indyk, P. [2002]. Approximate clustering via core-sets, *Proc. Symposium on Theory of Computing*, ACM, pp. 250–257.
- Bajaj, C. [1988]. The algebraic degree of geometric optimization problems, *Discrete & Computational Geometry* **3**(2): 177–191.
- Barequet, G. and Papadopoulou, E. [2013]. On the farthest-neighbor Voronoi diagram of segments in three dimensions, *10th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, IEEE, pp. 31–36.
- Barequet, G. and Papadopoulou, E. [2014]. On farthest-site Voronoi diagrams of line segments and lines in three and higher dimensions, *30th European Workshop on Computational Geometry* .
- Barequet, G., Papadopoulou, E. and Suderland, M. [2019]. Unbounded regions of high-order Voronoi diagrams of lines and segments in higher dimensions, *30th International Symposium on Algorithms and Computation*, Vol. 149 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 62:1–62:15.
- Beck, A. and Sabach, S. [2015]. Weiszfeld’s method: Old and new results, *Journal of Optimization Theory and Applications* **164**(1): 1–40.
- Beck, J. [1981]. Balanced two-colorings of finite sets in the square I, *Combinatorica* **1**(4): 327–335.
- Bennett, H., Papadopoulou, E. and Yap, C. [2016]. Planar minimization diagrams via subdivision with applications to anisotropic Voronoi diagrams, *Computer Graphics Forum* **35**(5): 229–247.
- Bennett, H. and Yap, C. [2017]. Amortized analysis of smooth quadtrees in all dimensions, *Computational Geometry* **63**: 20–39.
- Bhattacharya, B. B. [2011]. On the Fermat-Weber point of a polygonal chain and its generalizations, *Fundamenta Informaticae* **107**(4): 331–343.
- Bohler, C., Cheilaris, P., Klein, R., Liu, C., Papadopoulou, E. and Zavershynskiy, M. [2015]. On the complexity of higher order abstract voronoi diagrams, *Comput. Geom.* **48**(8): 539–551.

- Boissonnat, J., Sharir, M., Tagansky, B. and Yvinec, M. [1998]. Voronoi diagrams in higher dimensions under certain polyhedral distance functions, *Discrete & Computational Geometry* **19**(4): 485–519.
- Bose, P., Maheshwari, A. and Morin, P. [2003]. Fast approximations for sums of distances, clustering and the Fermat-Weber problem, *Computational Geometry* **24**(3): 135–146.
- Brouwer, L. E. J. [1911]. Über Abbildung von Mannigfaltigkeiten, *Mathematische Annalen* **71**(1): 97–115.
- Burr, M. A. [2016]. Continuous amortization and extensions: With applications to bisection-based root isolation, *Journal of Symbolic Computation* **77**: 78–126.
- Burr, M., Krahmer, F. and Yap, C. [2009]. Continuous amortization: A non-probabilistic adaptive analysis technique, *Electronic Colloquium on Computational Complexity* **TR09**(136).
- Carmi, P., Har-Peled, S. and Katz, M. J. [2005]. On the Fermat-Weber center of a convex object, *Computational Geometry* **32**(3): 188–195.
- Chazelle, B. [1991]. An optimal convex hull algorithm and new results on cuttings (extended abstract), *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico*, IEEE Computer Society, pp. 29–38.
- Chazelle, B. [1993]. An optimal convex hull algorithm in any fixed dimension, *Discrete & Computational Geometry* **10**: 377–409.
- Chazelle, B. and Edelsbrunner, H. [1985]. An improved algorithm for constructing k th-order voronoi diagrams, in J. O'Rourke (ed.), *Proceedings of the First Annual Symposium on Computational Geometry, Baltimore, Maryland, USA, June 5-7, 1985*, ACM, pp. 228–234.
- Cheong, O., Everett, H., Glisse, M., Gudmundsson, J., Hornus, S., Lazard, S., Lee, M. and Na, H. [2011]. Farthest-polygon Voronoi diagrams, *Computational Geometry: Theory and Applications* **44**(4): 234–247.
- Chew, L. P., Kedem, K., Sharir, M., Tagansky, B. and Welzl, E. [1998]. Voronoi diagrams of lines in 3-space under polyhedral convex distance functions, *J. Algorithms* **29**(2): 238–255.

- Chin, H. H., Madry, A., Miller, G. L. and Peng, R. [2013]. Runtime guarantees for regression problems, *Proc. Innovations in Theoretical Computer Science*, ACM, pp. 269–282.
- Chiu, M.-K., Korman, M., Suderland, M. and Tokuyama, T. [2022]. Distance bounds for high dimensional consistent digital rays and 2-d partially-consistent digital rays, *Discrete & Computational Geometry* pp. 1–43.
- Chiu, M. and Korman, M. [2018]. High dimensional consistent digital segments, *SIAM Journal on Discrete Mathematics* **32**(4): 2566–2590.
- Chowdhury, I. and Gibson, M. [2015]. A characterization of consistent digital line segments in \mathbb{Z}^2 , in N. Bansal and I. Finocchi (eds), *Proceedings of the 23rd Annual European Symposium on Algorithms*, Vol. 9294, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 337–348.
- Chowdhury, I. and Gibson, M. [2016]. Constructing consistent digital line segments, in E. Kranakis, G. Navarro and E. Chávez (eds), *Proceedings of the 12th Latin American Theoretical Informatics Symposium*, Vol. 9644, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 263–274.
- Christ, T., Pálvölgyi, D. and Stojaković, M. [2012]. Consistent digital line segments, *Discrete & Computational Geometry* **47**(4): 691–710.
- Chun, J., Korman, M., Nöllenburg, M. and Tokuyama, T. [2009]. Consistent digital rays, *Discrete and Computational Geometry* **42**(3): 359–378.
- Cieslik, D. [2013]. *Steiner minimal trees*, Vol. 23, Springer Science & Business Media.
- Clarkson, K. L. [1987]. New applications of random sampling in computational geometry, *Discret. Comput. Geom.* **2**: 195–222.
- Clarkson, K. L. and Shor, P. W. [1989]. Applications of random sampling in computational geometry, II, *Discrete & Computational Geometry* **4**(5): 387–421.
- Cockayne, E. J. and Melzak, Z. A. [1969]. Euclidean constructibility in graph-minimization problems, *Mathematics Magazine* **42**(4): 206–208.
- Cohen, M. B., Lee, Y. T., Miller, G. L., Pachocki, J. and Sidford, A. [2016]. Geometric median in nearly linear time, *Proc. Symposium on Theory of Computing*, ACM, pp. 9–21.

- Das, S. and Sarvottamananda, S. [2018]. Computing the Minkowski sum of convex polytopes in \mathbb{R}^d , *CoRR* **abs/1811.05812**.
- Dekker, T. J. [1967]. Finding a zero by means of successive linear interpolation, *Constructive Aspects of the Fundamental Theorem of Algebra*, Wiley Interscience, pp. 37–48.
- Dobkin, D. P., Gunopulos, D. and Maass, W. [1996]. Computing the maximum bichromatic discrepancy, with applications to computer graphics and machine learning, *Journal of Computer and System Sciences* **52**(3): 453–470.
- Dumitrescu, A., Jiang, M. and Tóth, C. D. [2011]. New bounds on the average distance from the Fermat-Weber center of a planar convex body, *Discrete Optimization* **8**(3): 417–427.
- Edelsbrunner, H., Guibas, L. J. and Sharir, M. [1989]. The upper envelope of piecewise linear functions: Algorithms and applications, *Discrete & Computational Geometry* **4**: 311–336.
- Edelsbrunner, H. and Seidel, R. [1986]. Voronoi diagrams and arrangements, *Discrete & Computational Geometry* **1**: 25–44.
- Edelsbrunner, H., Seidel, R. and Sharir, M. [1993]. On the zone theorem for hyperplane arrangements, *SIAM Journal on Computing* **22**(2): 418–429.
- Emiris, I. Z., Tsigaridas, E. P. and Tzoumas, G. M. [2006]. The predicates for the Voronoi diagram of ellipses, *Proc. Symposium on Computational Geometry*, ACM, pp. 227–236.
- Everett, H., Lazard, D., Lazard, S. and Din, M. S. E. [2009]. The Voronoi diagram of three lines, *Discrete & Computational Geometry* **42**(1): 94–130.
- Fagnano, G. F. [1775]. Problemata quaedam ad methodum maximorum et minimorum spectantia, *Nova Acta Eruditorum* pp. 281–303.
- Fekete, S. P., Mitchell, J. S. and Beurer, K. [2005]. On the continuous Fermat-Weber problem, *Operations Research* **53**(1): 61–76.
- Feldman, D. and Langberg, M. [2011]. A unified framework for approximating and clustering data, *Proc. Symposium on Theory of Computing*, ACM, pp. 569–578.
- Gibson-Lopez, M. and Zamarripa, S. [2022]. Optimal bounds for weak consistent digital rays in 2d, *arXiv preprint arXiv:2205.03450* .

- Graham, R. L. [1972]. An efficient algorithm for determining the convex hull of a finite planar set, *Inf. Process. Lett.* **1**(4): 132–133.
- Halperin, D. and Sharir, M. [2017]. Arrangements, *Handbook of Discrete and Computational geometry, third edition* pp. 723–762.
- Hamacher, H. and Drezner, Z. [2002]. Facility location: applications and theory, *Science & Business Media: Springer* .
- Hansen, E. R. [2006]. A multidimensional interval newton method, *Reliable Computing* **12**(4): 253–272.
- Hansen, E. R. and Sengupta, S. [1981]. Bounding solutions of systems of equations using interval analysis, *BIT* **21**: 203–211.
- Har-Peled, S. and Kushal, A. [2007]. Smaller coresets for k-median and k-means clustering, *Discrete & Computational Geometry* **37**(1): 3–19.
- Har-Peled, S. and Mazumdar, S. [2004]. On coresets for k-means and k-median clustering, *Proc. 36th Annual ACM Symposium on Theory of computing*, ACM, pp. 291–300.
- Hemmer, M., Setter, O. and Halperin, D. [2010]. Constructing the exact Voronoi diagram of arbitrary lines in three-dimensional space - with fast point-location, in M. de Berg and U. Meyer (eds), *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, Vol. 6346 of *Lecture Notes in Computer Science*, Springer, pp. 398–409.
- Huttenlocher, D. P., Kedem, K. and Sharir, M. [1993]. The upper envelope of Voronoi surfaces and its applications, *Discrete & Computational Geometry* **9**(3): 267–291.
- Icking, C. and Ma, L. [2001]. A tight bound for the complexity of Voronoi diagrams under polyhedral convex distance functions in 3D, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pp. 316–321.
- Jarvis, R. A. [1973]. On the identification of the convex hull of a finite set of points in the plane, *Inf. Process. Lett.* **2**(1): 18–21.
- Karavelas, M. I., Seidel, R. and Tzanaki, E. [2013]. Convex hulls of spheres and convex hulls of disjoint convex polytopes, *Computational Geometry: Theory and Applications* **46**(6): 615–630.

- Katz, I. N. [1974]. Local convergence in Fermat's problem, *Mathematical Programming* **6**(1): 89–104.
- Klee, V. [1980]. On the complexity of d-dimensional Voronoi diagrams, *Archiv der Mathematik* **34**(1): 75–80.
- Klein, R. [2016]. *Abstract Voronoi Diagrams*, Springer New York, pp. 5–8.
- Koltun, V. and Sharir, M. [2002]. Three dimensional Euclidean Voronoi diagrams of lines with a fixed number of orientations, in F. Hurtado, V. Sacristán, C. Bajaj and S. Suri (eds), *Proceedings of the 18th Annual Symposium on Computational Geometry, Barcelona, Spain, June 5-7, 2002*, ACM, pp. 217–226.
- Koltun, V. and Sharir, M. [2004]. Polyhedral Voronoi diagrams of polyhedra in three dimensions, *Discrete & Computational Geometry* **31**(1): 83–124.
- Krarpup, J. and Vajda, S. [1997]. On Torricelli's geometrical solution to a problem of Fermat, *Journal of Management Mathematics* **8**(3): 215–224.
- Krawczyk, R. [1969]. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, *Computing* **4**(3): 187–201.
- Kuhn, H. W. [1973]. A note on Fermat's problem, *Mathematical programming* **4**(1): 98–107.
- Kuratowski, K. [1966]. *Topology*, Academic Press.
- Lee, D. [1982]. On k -nearest neighbor voronoi diagrams in the plane, *IEEE Trans. Computers* **31**(6): 478–487.
- Lin, L. and Yap, C. [2011]. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach, *Discrete & Computational Geometry* **45**(4): 760–795.
- Lin, L., Yap, C. and Yu, J. [2012]. Non-local isotopic approximation of nonsingular surfaces, *Computer-Aided Design* **45**(2): 451–462.
- Luby, M. G. [1987]. Grid geometries which preserve properties of Euclidean geometry: A study of graphics line drawing algorithms., *NATO Conference on Graphics/CAD*, pp. 397–432.
- Mantas, I., Papadopoulou, E., Sacristán, V. and Silveira, R. I. [2021]. Farthest color voronoi diagrams: Complexity and algorithms, *Latin American Symposium on Theoretical Informatics*, Springer, pp. 283–295.

- Matoušek, J. [1999]. *Geometric Discrepancy: An Illustrated Guide*, Algorithms and Combinatorics, Springer Berlin Heidelberg.
URL: <https://books.google.co.jp/books?id=BKvXj1GisPOC>
- McMullen, P. [1970]. The maximum numbers of faces of a convex polytope, *Mathematika* **17**(2): 179–184.
- Mello, L. F. and dos Santos, L. R. [2018]. On the location of the minimum point in the Euclidean distance sum problem, *São Paulo Journal of Mathematical Sciences* **12**: 108–120.
- Mitchell, J. S. B. and O’Rourke, J. [2001]. Computational geometry column 42, *International Journal of Computational Geometry & Applications* **11**(5): 573–582.
- Moore, R. E. [1966]. *Interval Analysis*, Vol. 4, Prentice-Hall Englewood Cliffs, NJ.
- Morrison, K. E. [2010]. The fedex problem, *The College Mathematics Journal* **41**(3): 222–232.
- Nagy, G. S. [1950]. Tschirnhaus’sche Eiflächen und Eikurven, *Acta Mathematica Academiae Scientiarum Hungarica* **1**(1): 36–45.
- Nam, N. M. [2013]. The Fermat-Torricelli problem in the light of convex analysis, *ArXiv e-prints* .
- Nickel, K. [1969]. Triplex-algol and applications, *Interner Bericht des Instituts für Informatik der Universität Karlsruhe* .
- Nickel, K. [1971]. On the Newton method in interval analysis, *Technical report*, Wisconsin University-Madison Mathematics Research Center.
- Nie, J., Parrilo, P. A. and Sturmfels, B. [2008]. Semidefinite representation of the k -ellipse, *Algorithms in algebraic geometry*, Springer, pp. 117–132.
- Okabe, A., Boots, B., Sugihara, K. and Chiu, S. N. [1992]. Spatial tessellations: concepts and applications of Voronoi diagrams. 1992, Chichester, UK .
- Okabe, A., Boots, B., Sugihara, K. and Chiu, S. N. [2009]. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams (2nd Edition)*, Vol. 501, John Wiley & Sons.
- Ostresh Jr, L. M. [1978]. Convergence and descent in the Fermat location problem, *Transportation Science* **12**(2): 153–164.

- Papadopoulou, E. [2004]. The Hausdorff Voronoi diagram of point clusters in the plane, *Algorithmica* **40**(2): 63–82.
- Papadopoulou, E. and Dey, S. K. [2013]. On the farthest line-segment Voronoi diagram, *Int. J. Comput. Geometry Appl.* **23**(6): 443–460.
- Papadopoulou, E. and Lee, D.-T. [2004]. The Hausdorff Voronoi diagram of polygonal objects: A divide and conquer approach, *International Journal of Computational Geometry & Applications* **14**(06): 421–452.
- Papadopoulou, E. and Zavershynskiy, M. [2016]. The higher-order Voronoi diagram of line segments, *Algorithmica* **74**(1): 415–439.
- Parrilo, P. A. and Sturmfels, B. [2003]. Minimizing polynomial functions, *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **60**: 83–99.
- Petrović, M., Banjac, B. and Malešević, B. [2014]. The geometry of trifocal curves with applications in architecture, urban and spatial planning, *Spatium* pp. 28–33.
- Plantinga, S. and Vegter, G. [2004]. Isotopic approximation of implicit curves and surfaces, *Proc. of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ACM, pp. 245–254.
- Ratschek, H. and Rokne, J. [1984]. *Computer methods for the range of functions*, Horwood.
- Reinelt, G. [1991]. TSPLIB - A traveling salesman problem library, *ORSA Journal on Computing* **3**(4): 376–384.
- Samet, H. [1990]. *The Design and Analysis of Spatial Data Structures*, Addison-Wesley.
- Schmidt, W. [1972]. Irregularities of distribution, vii, *Acta Arithmetica* **21**(1): 45–50.
- Seidel, R. [1987]. On the number of faces in higher-dimensional Voronoi diagrams, in D. Soule (ed.), *Proceedings of the Third Annual Symposium on Computational Geometry, Waterloo, Ontario, Canada, 1987*, ACM, pp. 181–185.
- Sekino, J. [1999]. n -ellipses and the minimum distance sum problem, *The American mathematical monthly* **106**(3): 193–202.

- Shamos, M. I. and Hoey, D. [1975]. Closest-point problems, *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, IEEE Computer Society, pp. 151–162.
- Sharir, M. [1994]. Almost tight upper bounds for lower envelopes in higher dimensions, *Discrete & Computational Geometry* **12**: 327–345.
- Shary, S. P [2004]. Krawczyk operator revised, *Novosibirsk, Institute of Computational Technologies, Rússia* .
- Stanley, R. P [2004]. An introduction to hyperplane arrangements, *Geometric combinatorics* **13**: 389–496.
- Sturm, R. [1884]. Über den Punkt kleinster Entfernungssumme von gegebenen Punkten., *Journal für die reine und angewandte Mathematik* **97**: 49–61.
- Tucker, W. [2011]. *Validated Numerics: A short intro to rigorous computations*, Princeton Press.
- von Tschirnhaus, E. W. [1695]. *Medicina Mentis Et Corporis*, Fritsch, Lipsiae.
- Wang, C., Chiang, Y.-J. and Yap, C. [2015]. On soft predicates in subdivision motion planning, *Computational Geometry: Theory and Applications*. **48**(8): 589–605.
- Weber, A. [1909]. Über den Standort der Industrien, *English translation by CJ Friedrich (1929) Theory of the Location of Industries* .
- Weiszfeld, E. [1937]. Sur le point pour lequel la somme des distances de n points donnés est minimum, *Tohoku Mathematical Journal, First Series* **43**: 355–386.
- Xu, J. and Yap, C. [2019]. Effective subdivision algorithm for isolating zeros of real systems of equations, with complexity analysis, *Proc. International Symposium on Symbolic and Algebraic Computation*, ACM, pp. 399–406.
- Xue, G. and Ye, Y. [1997]. An efficient algorithm for minimizing a sum of Euclidean norms with applications, *SIAM Journal on Optimization* **7**(4): 1017–1036.