

COMPUTATION AND RECOGNITION OF

# WEIGHTED SKELETAL STRUCTURES

IN THE PLANE

Günther Eder

Cumulative dissertation submitted to the Faculty of Natural Sciences of the University of Salzburg in partial fulfillment of the requirements for the doctoral degree Dr. techn.

Abridged version.

Supervisor: Martin Held  
Department of Computer Sciences  
University of Salzburg

**Department of Computer Sciences**  
University of Salzburg  
Jakob-Haringer-Straße 2  
5020 Salzburg  
Austria

**Günther Eder**  
*Computation and Recognition of Weighted Skeletal Structures in the Plane*  
April 2019

[version 1.0 (abridged) · commit f70dfd ]

## ABSTRACT

---

We are given input embedded in the plane that is in the form of either a simple polygon, a planar straight-line graph, or a set of points, rectangles, or line-segments. Every pair of input elements defines a *bisector*, that is the set of points that are equidistant to both input elements, in respect to a predefined distance function.

In this cumulative dissertation the focus lies on geometric straight-line structures that are defined as portions of such bisectors. Among others, we investigate two well-known structures, namely the *straight skeleton* and the *Voronoi diagram*. The latter is explored by defining a weighted variant that still results in a straight-line structure. The straight skeleton is researched in a weighted setting, and under restriction of the input, where for both settings new approaches are introduced. The straight skeleton inside of a simple polygon can be used to erect a z-monotone terrain called roof. Generalizing from such a straight skeleton induced roof we introduce a new roof type. As part of this cumulative research we discuss certain properties of this new roof type and show how to compute it. Additionally, we view the straight skeleton in the sense of a recognition and reconstruction problem.



## ACKNOWLEDGMENTS

---

I thank my advisor Martin Held for the guidance, patience, and support he provided over the last years. I thank my friend and colleague Peter Palfrader for the numerous discussions we had related to our field. I especially thank my wife and family for their support.

Work on this thesis has been supported by the Austrian Science Fund (FWF) under grant number P25816 – N15 and ORD 53 – VO.



## CONTENTS

---

I	INTRODUCTION . . . . .	1
1	Preliminaries . . . . .	3
1.1	Straight Skeleton . . . . .	3
1.1.1	Bisector Graph & Roof Model . . . . .	4
1.1.2	Weighted Straight Skeleton . . . . .	6
1.1.3	Ambiguities . . . . .	6
1.1.4	Applications . . . . .	7
1.1.5	Computation . . . . .	9
1.2	Voronoi Diagram . . . . .	10
1.2.1	Weighted Voronoi Diagram . . . . .	11
1.2.2	Applications . . . . .	12
1.2.3	Computation . . . . .	12
2	Contribution . . . . .	15
2.1	Minimum & Maximum Volume Roofs . . . . .	15
2.2	Straight Skeleton of an Orthogonal Monotone Polygon . . . . .	16
2.3	Weighted Straight Skeletons and a Line Arrangement . . . . .	17
2.4	Recognizing Weighted Straight Skeletons . . . . .	18
2.5	Weighted Voronoi Diagrams in the Maximum-Norm . . . . .	19
	BIBLIOGRAPHY . . . . .	21
II	PUBLICATIONS (ABRIDGED) . . . . .	27
	Min-/Max-Volume Roofs of Polygonal Footprints of Buildings . . . . .	28
	Straight Skeleton of an Orthogonal Monotone Polygon . . . . .	30
	Positively Weighted Straight Skeletons of Simple Polygons . . . . .	32
	Recognizing Geometric Trees as Weighted Straight Skeletons . . . . .	34
	Multiplicatively Weighted Voronoi Diagrams in the Maximum Norm . . . . .	36



## Part I

# INTRODUCTION



In [Chapter 1](#) of my cumulative dissertation I give a short overview about the relevant geometric structures. In the subsequent sections the basic terminology is introduced, we see where the relevant structures originate, and look at current state of the art algorithms for their computation.

We start with a closer look at the *straight skeleton* in [Section 1.1](#), then in [Section 1.2](#), we discuss the famous *Voronoi diagram*. We survey selected publications on these structures and discuss how to construct them and where they are applied.

In [Chapter 2](#), the contribution of this cumulative research is summarized. This includes a three-dimensional roof structure that can be constructed on top of a simple polygon. Such a roof can be raised in combination with, for example, the induced straight skeleton. Then, we look at the straight skeleton, first by restricting its input, then in a weighted setting, and finally in the sense of a recognition problem. To complete the outline, we change the focus to the Voronoi diagram. We inspect the Voronoi diagram of weighted points, rectangles, and line-segments under the maximum norm.

The main part of my dissertation consists of already published and submitted papers to peer-reviewed journals and workshops. These papers are cumulated in [Part II](#).

## 1.1 STRAIGHT SKELETON

The *straight skeleton* was first seen in the work of Peschka from 1877 [[Pes77](#)]. Later, Aichholzer and Aurenhammer rediscover the straight skeleton and introduce it formally [[AAAG95](#)]. Both referenced straight skeletons are illustrated in [Figure 1](#). The straight skeleton can be constructed inside of a simple polygon  $\mathcal{P}$  embedded in the plane and forms a geometric tree that terminates at the vertices of  $\mathcal{P}$ . It forms a straight-line structure that is of linear size in respect to the size of  $\mathcal{P}$ . Furthermore, the straight skeleton decomposes the polygon into a linear number of faces, one for each polygon edge. One can also compute the straight skeleton on the outside of  $\mathcal{P}$  as well as on top of a planar straight-line graph [[AA96](#)].

The *wavefront propagation* is a well-known strategy for computing straight skeletons [[AAAG95](#), [HH12](#), [Bie+15b](#)]. At the time the propagation starts the wavefront is equal to  $\mathcal{P}$ . As time progresses the propagation process simulates a shrinking of the wavefront. In this process every wavefront edge moves at unit speed and in a self-parallel manner into the interior of the polygon while maintaining a closed boundary. The vertices of this shrinking wavefront trace

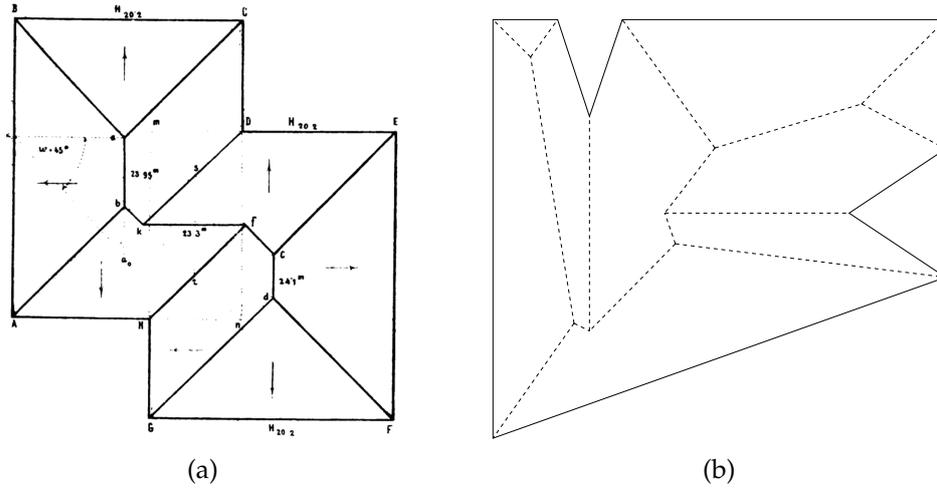


Figure 1: (a) Figure 75 of Peschka’s work from 1877 [Pes77], the interior structure of the axis-aligned polygon features the straight skeleton; (b) Figure 1 from the work of Aichholzer and Aurenhammer [AAAG95], the straight skeleton in dashed lines of a simple polygon.

out arcs which form the straight skeleton. In Figure 2a, the straight skeleton (blue) of a simple polygon is illustrated as well as the wavefront (gray) at distinct times. To maintain the weak planarity of the wavefront two event types have to be handled: edge events and split events. An edge event occurs when a wavefront edge shrinks to zero length. A split event occurs when a reflex wavefront vertex (interior angle is larger than  $\pi$ ) crashes into the interior of an opposing wavefront edge. These two types of events produce the (interior) nodes of the straight skeleton such that at least three arcs meet in a common node. Additionally, there is a node of degree one for every vertex of  $\mathcal{P}$ .

1.1.1 Bisector Graph & Roof Model

In the work of Aichholzer and Aurenhammer [AAAG95] not only the straight skeleton is defined but also both the *bisector graph* and the *roof model*. A graph is a bisector graph if every arc (i) is formed from a bisector portion of two edges of  $\mathcal{P}$ , and (ii) is bounded by a degree one vertex of  $\mathcal{P}$  or a degree at least three node of the bisector graph.

Let  $\ell_i, \ell_j$  denote the supporting lines of two distinct edges of  $\mathcal{P}$  with interior sides inherited from their defining edges. Then  $\ell_i, \ell_j$  define a unique bisector line by propagating both lines to their common interior (exterior) side: The intersection of the two moving lines  $\ell_i, \ell_j$  traces out their bisector. The case of parallel edges is discussed in Section 1.1.3.

The roof model can be seen as a half-plane arrangement rooted at the edges of  $\mathcal{P}$  in three-space: We place a half-plane  $\Pi_e$  originating on the supporting line

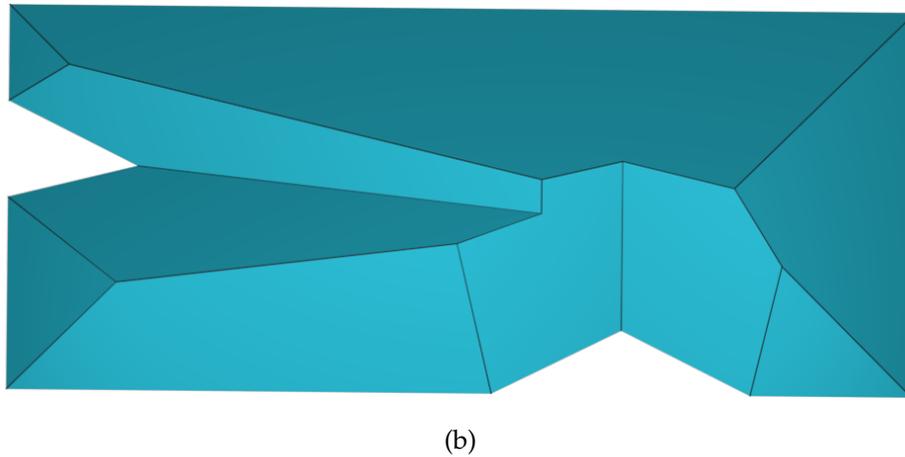
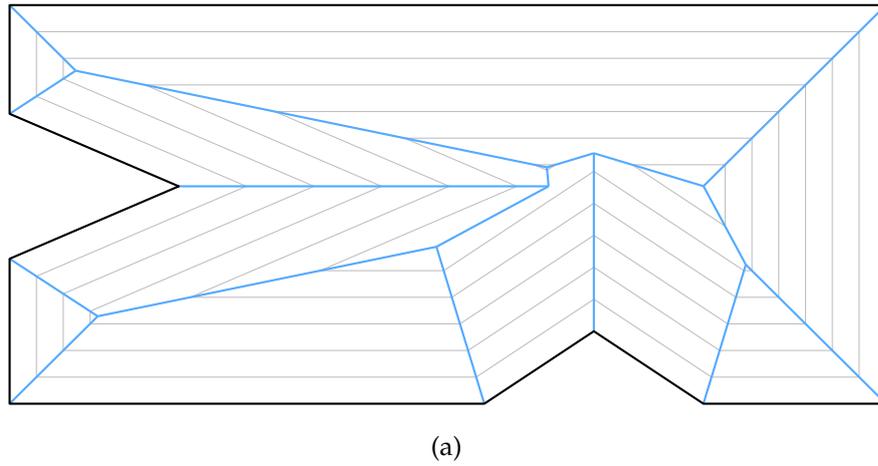


Figure 2: (a) A simple polygon (black), the wavefront at fixed times (gray), and the straight skeleton (blue); (b) The respective roof by lifting the straight skeleton shown in (a) into  $\mathbb{R}^3$ .

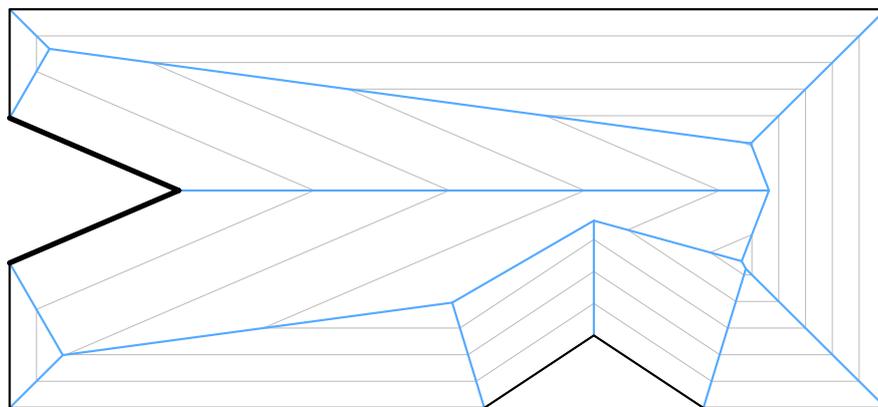


Figure 3: The weighted straight skeleton of the same input as in [Figure 2](#) where the weight of both two thick input edges is higher than the weight of each of the remaining boundary edges.

of every edge  $e$  of  $\mathcal{P}$ , such that  $\Pi_e$  is inclined towards the interior of  $e$  at a fixed dihedral angle. Then, the intersection  $\Pi_e \cap \Pi_{e'}$  forms a ray, assuming  $e$  and  $e'$  are not parallel. If we project that ray orthogonally onto the  $xy$ -plane, then it co-insides with the bisector of the corresponding edges of  $\mathcal{P}$ .

Aichholzer and Aurenhammer formulate this bijection between bisector graph and roof model. That is, for every roof we can map its edges orthogonally onto the  $xy$ -plane and obtain a bisector graph, and vice versa, in respect to a fixed  $\mathcal{P}$ . Since the straight skeleton is a bisector graph, we can raise it into three-space and form a roof over  $\mathcal{P}$ . That roof forms a  $z$ -monotone terrain that covers  $\mathcal{P}$  and is incident on the boundary of  $\mathcal{P}$ . Additionally, such a straight skeleton induced roof drains water to the boundary of  $\mathcal{P}$  [AAAG95]. Thus, this roof contains no sink, that is, a point on the roof that forms a local minimum but is not part of the boundary of  $\mathcal{P}$ . An illustration of a straight skeleton and its erected roof can be seen in Figures 2a and 2b.

### 1.1.2 Weighted Straight Skeleton

In the weighted scenario every input edge  $e$  of  $\mathcal{P}$  requires an additional parameter. This parameter provides a per edge weight and is part of the input. In general, there are two weighted variants: *additively* and *multiplicatively* weighted straight skeletons. In the former case, a wavefront edge still moves with unit speed but starts at the time given by the additive weight. In the latter case, a wavefront edge moves with a speed given by the multiplicative weight. Note, for convenience we use the term weighted straight skeleton by which we refer to the multiplicative variant. See an example of a weighted straight skeleton in Figure 3.

Biedl et al. [Bie+15b] show that many properties of unweighted straight skeletons are preserved for positively weighted straight skeletons of simple polygons. In particular the straight skeleton is connected, forms a tree, has no crossings, and consists of  $n + v - 1$  arcs, where  $n$  denotes the size of the polygon and  $v$  denotes the number of straight skeleton nodes.

### 1.1.3 Ambiguities

In general, the wavefront traces out the straight skeleton over time. This is also true for the weighted variant. That would imply that the straight skeleton is not only a tree but also directed. However, as soon as parallel edges are present in  $\mathcal{P}$  this is no longer true. Consider two wavefront edges that are parallel and move toward each other. Their common bisector is defined as the line that coincides with the intersection at the time the two wavefront edges meet. Clearly this intersection does not induce a direction for the formed arc.

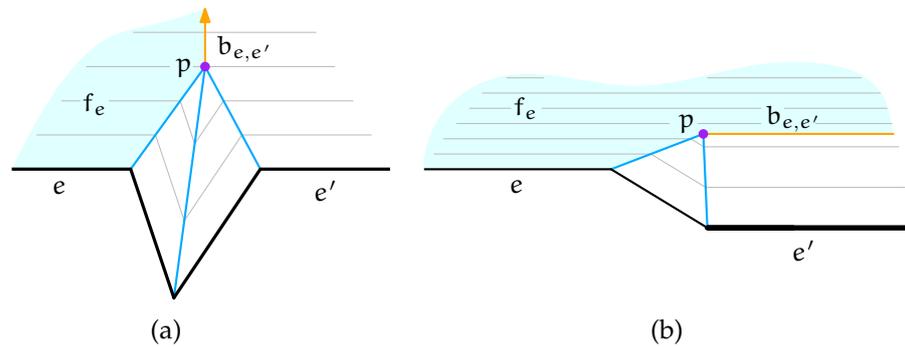


Figure 4: (a,b) Two parallel wavefront edges of  $e$  and  $e'$  become adjacent at a point  $p$ , the face  $f_e$  incident at  $e$  is highlighted; In (a), the common bisector  $b_{e,e'}$  (orange) is oriented perpendicular to  $e, e'$ ; In (b),  $e$  has lower weight than  $e'$ , we let the bisector  $b_{e,e'}$  (orange) terminate the wavefront edge of  $e'$ .

Let us assume that two wavefront edges move in the same direction with equal weights and meet at some point  $p$ , then one can define the resulting bisector as directed perpendicular to them originating at  $p$ . In this way the resulting straight skeleton stays connected and every face is incident at a unique input edge, as visualized in Figure 4a. However, if these wavefront edges have different weights and become adjacent, then the direction of their bisector is undefined. One feasible approach is to always pick the edge with lower weight and thereby terminate the wavefront edge with higher weight [Bie+15b]. An example of this scenario is illustrated in Figure 4b.

#### 1.1.4 Applications

The straight skeleton provides an alternative to the *medial axis*: The medial axis was introduced by Blum in 1967 [Blu67] and is defined inside a simple polygon  $\mathcal{P}$  as the set of all points that have more than one closest point on the boundary of  $\mathcal{P}$ . Where the medial axis in general contains arcs of parabolas, the straight skeleton is always formed out of straight-line segments [AAAG95]. Hence, the straight skeleton has a lower algebraic degree, which can improve the robustness for applications that build on the straight skeleton.

Polygon offsetting is the task of constructing an offset curve inside or outside of a simple polygon  $\mathcal{P}$ . This offset curve has a specific distance to the boundary of  $\mathcal{P}$ . Here, the straight skeleton can be applied in order to compute a mitered offset of  $\mathcal{P}$  [PH15]. Given the straight skeleton the offset computation is straight forward as it can be constructed by traversing the straight skeleton face by face. Simply walk along the boundary of a face and embed the offset edge before iterating to the next face.

As described in Section 1.1.1, the straight skeleton can be used to raise roofs that drain water. In 3D urban modeling this property is applied to automate roof

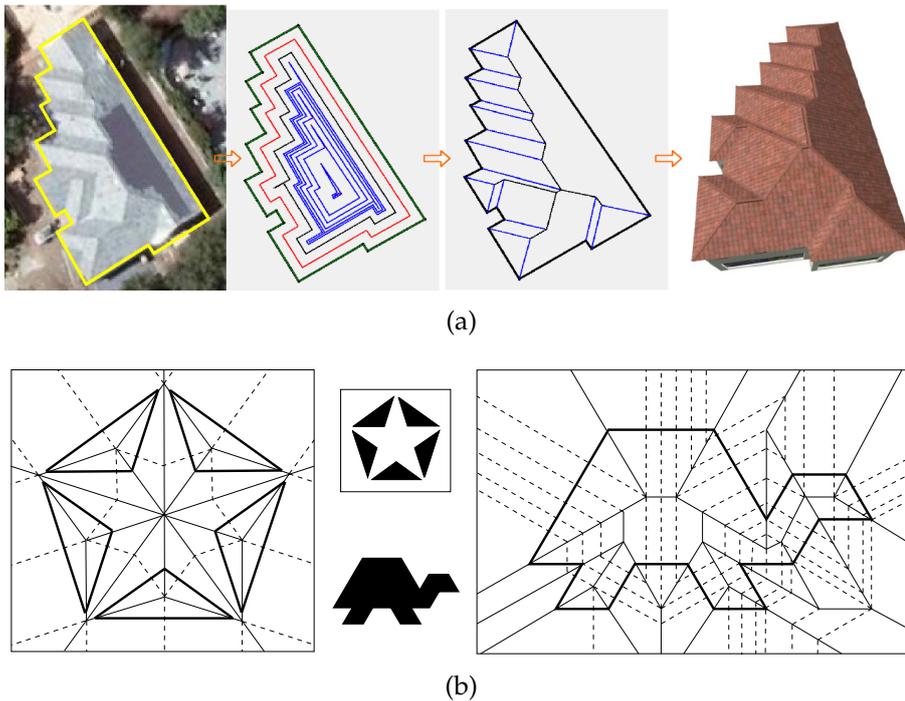


Figure 5: (a) An illustration from the work of Sugihara and Khmelevsky, illustrating the process of automated roof construction [SK18].  
 (b) An example from Demaine et al. showing two models and their fold-and-cut graph [DDL99].

computation for building footprints [LD03, SK18, Hub12]. In the work of Sugihara and Khmelevsky [SK18] building footprints are recognized through image processing. Then, these footprints are used as input polygons for the straight skeleton computation. An illustration of their work is shown in Figure 5a. A similar approach can be applied to cartographic modeling, in particular for automatic river bed construction [Hub12].

Tomoeda and Sugihara [TS12] apply the straight skeleton to create an illusionary 3D effect on a polygonal structure in the plane. They aim to enhance perception of name boards and signs using this illusionary depth effect.

Another interesting area of application was investigated by Demain et al., they use the straight skeleton for solving mathematical origami [DDL99]. The fold-and-cut theorem states that any polygonal shape can be cut out of a piece of paper with a single straight cut after a precise folding. In Figure 5b two examples of his work are illustrated.

In the work of Hauner and Sester [HS08] the straight skeleton is utilized to automate cartographic computation. They collapse polygonal regions and represent them by the induced straight skeleton. For example, they collapse the area of a road on a map to its center line.

## 1.1.5 Computation

Contradicting common intuition, the straight skeleton cannot be obtained by computing the lower envelope of a simple three-space arrangement. The difficulty in computing the straight skeleton hinges on the determination of the split events. During the propagation a split event is caused by a reflex wavefront vertex  $v$  that crashes into an opposing wavefront edge. Determining at what time  $v$  crashes in which edge is difficult as the wavefront changes as time progresses [AAAG95].

Aichholzer and Aurenhammer [AAAG95] derive an algorithm from their definition. The wavefront propagation, as described above, can simulate all events and output the straight skeleton. Either compute all events that might take place and store them in a priority queue sorted by event time, which overall requires  $\mathcal{O}(n^2 \log n)$  time and  $\mathcal{O}(n^2)$  space, or use  $\mathcal{O}(n^3)$  time but only linear space.

Aichholzer and Aurenhammer also introduce a triangulation-based approach which is investigated in detail and implemented by Palfrader et al. [AA96, PHH12]. By maintaining a kinetic triangulation of the shrinking wavefront they identify the collapse times of the moving triangles. Then, the straight skeleton events only occur at such triangle collapses. The best known worst case time bound is  $\mathcal{O}(n^3 \log n)$ . However, Palfrader et al. observe that their implementation takes  $\mathcal{O}(n \log n)$  time for all practical inputs and linear memory consumption.

In 1999, Eppstein and Erickson [EE99] introduce the first sub-quadratic algorithm which takes  $\mathcal{O}(n^{17/11+\varepsilon})$  time and space, where  $\varepsilon > 0$ . They raise triangles originating at the input edges into three-space and then apply a 3D ray shooting query to obtain the closest triangle to a certain ray. These rays are induced by the reflex input vertices. They achieve a sub-linear query time by employing a technique for maintaining closest pairs. Their approach also works to construct the positively weighted straight skeleton in  $\mathcal{O}(n^{8/5+\varepsilon})$  time and space. For unrestricted input polygons their result is still the best-known for both the weighted and unweighted setting.

In the same work they also introduce the *motorcycle graph*: A motorcycle is defined as a point in the plane that has a given direction and velocity. Suppose we have  $n$  such motorcycles that start at the same time. As a motorcycle moves it leaves a trace behind and it only stops if it runs into another motorcycle's trace. The motorcycle graph is then the set of these  $n$  traces after infinite time. Knowledge of the motorcycle graph helps in computing the straight skeleton more efficiently.

In 2002, Cheng and Vigneron [CV02] introduce a new approach to compute the motorcycle graph in  $\mathcal{O}(n\sqrt{n} \log n)$  time. Their algorithm uses  $\varepsilon$ -nets to construct a  $1/\sqrt{\pi}$ -cutting of the input [Cha93]. These cuttings can be computed efficiently and enable Cheng and Vigneron to construct the motorcycle graph

cell by cell. Using the motorcycle graph they construct the straight skeleton in  $\mathcal{O}(n \log^2 n + r\sqrt{r} \log r)$  expected time. Later, Cheng et al. [CMV16] improve this bound to  $\mathcal{O}(n(\log n) \log r + r^{4/3+\epsilon})$ . Note that these algorithms are only applicable if the input is in general position: At most three input edges are tangent to a common circle.

In 2015, Biedl et al. show how to compute the straight skeleton of a monotone polygon in  $\mathcal{O}(n \log n)$  time [Bie+15a]. They first construct the straight skeleton of each monotone chain and then merge both skeletons in a second step. The key insight for computing the straight skeleton of a single monotone chain is, that every combinatorial change in the wavefront is observed by an edge event. Therefore, no split event must be identified which simplifies the computation considerably.

## 1.2 VORONOI DIAGRAM

The Voronoi diagram is named after the famous mathematician Georgy Feodosevich Voronoi who introduced the diagram in 1908 [Voro8]. Note, that the diagram is often referred to as Dirichlet tessellation, after Lejeune Dirichlet due to his work from 1850 [Dir50]. Informally, the diagram was first seen as early as 1644, in a work of Descartes [Des44]. See an illustration of his book in Figure 6a. In 1970, Shamos and Hoey [SH75] introduced the Voronoi diagram into the field of computational geometry.

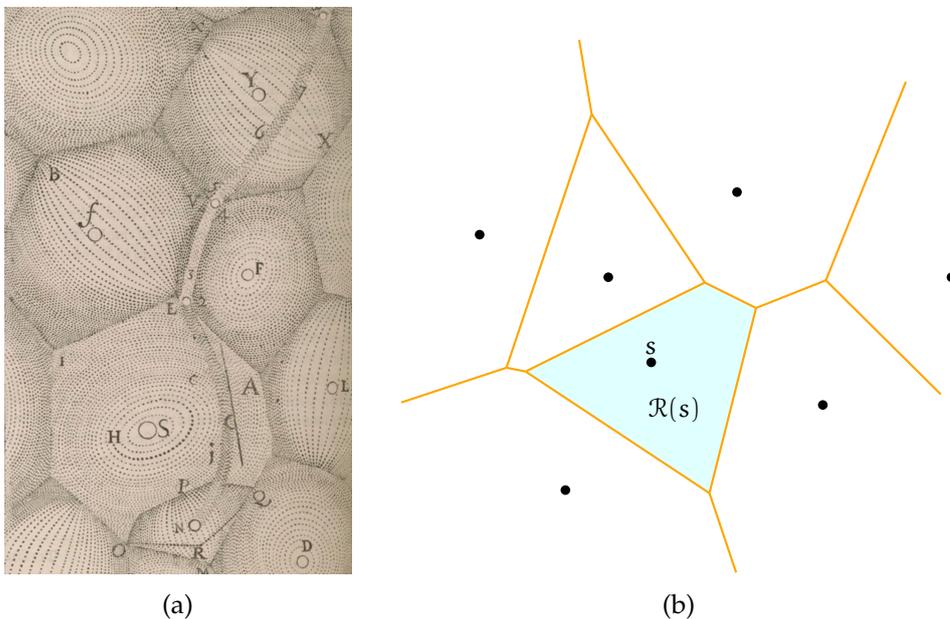


Figure 6: (a) An illustration of the first known Voronoi like structure in the work of René Descartes [Des44]; (b) The Voronoi diagram (orange) of a set of points, a single highlighted region  $\mathcal{R}(s)$  of one site  $s$  of the point set.

Let  $S$  be a set of  $n$  points, *sites*, in the plane. Then, the Voronoi diagram  $\mathcal{V}(S)$  is a tessellation of the plane into  $n$  Voronoi regions. The Voronoi region  $\mathcal{R}(s)$  is formed by every point  $p$  that is closer or equidistant to  $s$  than to any other site in  $S$ , as stated in [Equation \(1\)](#).

$$\mathcal{R}(s) := \{p \in \mathbb{R}^2 : d(p, s) \leq d(p, s') \text{ for all } s' \in S \text{ where } s \neq s'\}. \quad (1)$$

The points on the region boundary are equidistant to at least two sites. The Voronoi diagram  $\mathcal{V}(S)$  is then the union of all region boundaries. In the classical setting the distance is measured in the Euclidean metric, i.e.,  $L_2$ -norm. See an illustration of a Voronoi diagram in [Figure 6b](#).

### 1.2.1 Weighted Voronoi Diagram

There are two well-known weighted Voronoi diagrams which apply either additive or multiplicative weights. Assuming  $L_2$ -norm, we can imagine expanding disks on every site of  $S$  that start with a radius of zero. Then, region boundaries are created when disks intersect on a point that was not already covered by another disk. In case of additive weights, all disks expand at unit speed but start expanding at different times, depending on the additive weight. If multiplicative weights are used, then every disk expands at a speed given by its weight. The weights are part of the input such that for every site  $s$  of  $S$  we are given a specific weight  $\sigma(s) \in \mathbb{R}^+$ .

In the following we will look at the multiplicatively weighted Voronoi diagram. Aurenhammer and Edelsbrunner define the weighted distance function  $d_w(p, s) = d(p, s)/\sigma(s)$ , between a point  $p$  in  $\mathbb{R}^2$  and a site  $s$  of  $S$  [[AE84](#)]. For two sites with distinct weights the bisector is defined by a circle that encloses the site with lower weight. Therefore, the weighted Voronoi diagram is composed of circular arcs. Aurenhammer and Edelsbrunner present a worst-case example for the combinatorial complexity of such a diagram which is quadratic with respect to the number of sites, depicted in [Figure 7](#).

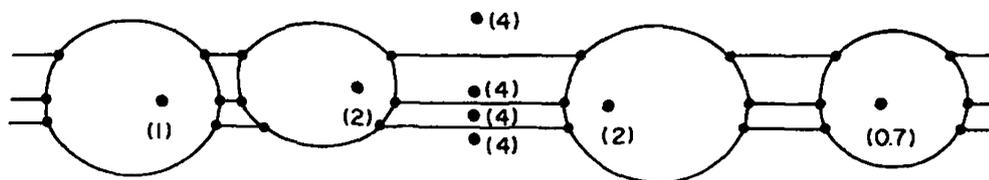


Figure 7: The worst-case example of the weighted Voronoi diagram presented by Aurenhammer and Edelsbrunner [[AE84](#)]. Site weights are indicated in brackets.

### 1.2.2 Applications

The Voronoi diagram is widely used and has applications in natural sciences, health, engineering, and of course computer science. In computational geometry a variety of problems can be solved efficiently provided that the Voronoi diagram is known: Nearest neighbor search, that is, given a set of points find the closest point next to a query point; Maximum empty circle, given a simple polygon, find the largest disk that fits in the interior of the polygon. Another application for the Voronoi diagram is in robot motion planning. A common task is to move a robot of specific shape from a given start to a given end point. On the area between the start and end point are some obstacles. The problem statement is twofold: (i) Can the robot reach the destination? (ii) Compute a path on which the robot can travel to its destination without colliding with an object [GMAMo6, Lat91].

To address some areas other than geometry, in biology Bock et al. [BTKA10] use the Voronoi diagram to model cell boundaries in cell formations. Sánchez-Gutiérrez et al. [Sán+16] apply the Voronoi diagram to better understand structure in muscle tissue in various conditions.

The research on Voronoi diagrams is comprehensive and further applications can be found in the book of Okabe [OBSC00].

### 1.2.3 Computation

In 1975, Shamos and Hoey [SH75] present the first deterministic algorithm to compute the Voronoi diagram in optimal  $\mathcal{O}(n \log n)$  time. They describe a divide & conquer approach where the set of points  $S$  is split recursively into a left and right half of roughly equal size. The Voronoi diagram is obtained by merging the solved diagrams of smaller size. The key insight is that the merge of two diagrams can be carried out in linear time.

In 1986, Fortune [For86] publishes his famous paper presenting a sweep line approach to compute the Voronoi diagram in  $\mathcal{O}(n \log n)$  time. Assume we have a plane,  $\mathbb{R}^2$ , containing all  $n$  sites and a line  $\ell$ , the sweep line. Then, we sweep over the plane from one side to the other by moving  $\ell$ , self-parallel, across the plane. During that sweep, every site that is intersected by  $\ell$  is processed. A classical sweep line algorithm guarantees a correct structure at any time in the already swept area. Fortune introduces a so called *beach line* that consists of parabolic arcs. Observe, the bisector between a point and a line forms such a parabolic arc. The beach line lies between  $\ell$  and the sites already covered by  $\ell$ . Then, Fortune shows that the Voronoi diagram is correct behind the beach line.

Later, Dehne and Klein [DK97] show how to apply a sweep line to obtain the Voronoi diagram in various metrics in the same time bound.

For the multiplicatively weighted Voronoi diagram, Aurenhammer and Edelsbrunner [AE84] introduce a worst case optimal approach in 1984. Their algorithm takes  $\mathcal{O}(n^2)$  time and space and applies geometric inversion. Roughly, they construct a unique sphere in three-space for every bisector between a fixed site  $s$  and every other site of  $S$ . Then, they use a spherical inversion and obtain a half-space for every sphere. By applying linear time half-space intersection, they are able to achieve this worst case optimal time bound.



In this chapter, I summarize the contribution of the work cumulated in [Part II](#). In the subsequent section we start with a closer look on a specific type of roof which builds on the roof model shown in [Section 1.1.1](#). We look at an algorithm that allows to find such a roof that binds the maximum and minimum volume on top of a given polygon. In [Section 2.2](#), our linear time algorithm to compute the straight skeleton of an orthogonal monotone polygon is sketched. Then, in [Section 2.3](#), we look at our algorithm to compute the weighted straight skeleton of a simple polygon, which aims to be easy to implement. In [Section 2.4](#), we discuss the reverse question: Given a weighted straight skeleton as a geometric graph, can we recognize it as such and if so, can we reconstruct a polygon that induces the given weighted straight skeleton? Finally, in [Section 2.5](#), we discuss an incremental construction approach to compute the weighted Voronoi diagram in the maximum norm.

## 2.1 MINIMUM & MAXIMUM VOLUME ROOFS

In [Section 1.1.1](#), we described the relationship between bisector graph and roof model. Aichholzer and Aurenhammer [[AAAG95](#)] show that the straight skeleton induced roof drains water: Assume a raindrop lands on a roof facet and takes the steepest descent, then it will always reach the boundary of the input polygon. Additionally, they state that this roof is not the maximum- or minimum-volume roof over a given polygon. We pick up this lead in *Min-/Max-Volume Roofs Induced by Bisector Graphs of Polygonal Footprints of Buildings* [[EHP18](#)] ([page 28](#)). Preliminary work was presented at the EuroCG workshop in 2016 [[EHP16](#)].

The basic idea is that the roof model, in general, allows for a variety of different  $z$ -monotone terrains over an input  $\mathcal{P}$ . We assume a fixed dihedral angle for every half-plane in the roof model. Then, the different terrains constructed using the roof model enclose different volumes.

We take a step back and observe which bisector graphs induce a roof that drains water. We define a restricted roof model and call it *natural roof*, which allows only for roofs that drain water. Then, we define additional events to allow for a wavefront like approach that traces out such a roof with minimum or maximum volume. We show that a greedy approach, that selects maximizing or minimizing events during the propagation, leads to the respective roof. Our proposed algorithm computes both roofs in  $\mathcal{O}(n^3 \log n)$  time and quadratic space. Furthermore, we establish a  $\Omega(2^{\Theta(n^2)})$  lower bound on the number of different natural roofs for a specific input.

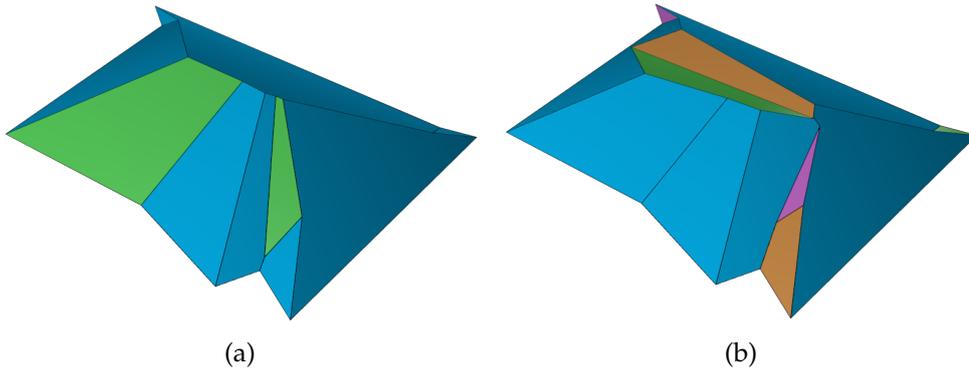


Figure 8: (a) Maximum-, (b) Minimum-Volume roof over a given input polygon.

Our approach is implemented in C++ using CGAL version 4.9 [CGAL]. The illustrated roofs in Figure 8 are computed with our implementation and rendered using Blender. The source code is freely available on github [RO]: <https://github.com/guenthereder/roofer>.

## 2.2 STRAIGHT SKELETON OF AN ORTHOGONAL MONOTONE POLYGON

In *Computing the Straight Skeleton of an Orthogonal Monotone Polygon in Linear Time* [EHP19a] (page 30), we present an optimal algorithm. Given an orthogonal monotone polygon  $\mathcal{P}$ , we split it into two monotone chains. Then, the straight skeleton of each chain is computed individually. A final merge of the two skeletons reveals the straight skeleton of  $\mathcal{P}$ .

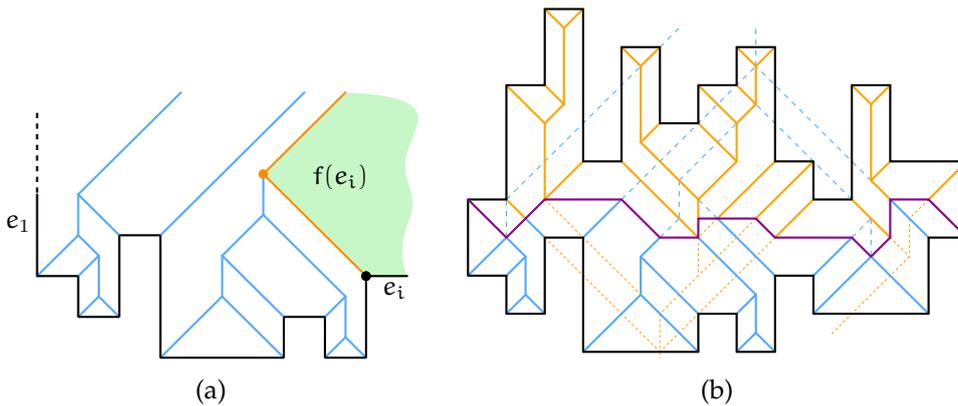


Figure 9: (a) Adding the left part (orange) of the highlighted face  $f(e_i)$  from the newly added edge  $e_i$ , then  $S^*$  consists of all interior arcs (blue and orange); (b) Merge of the straight skeletons of the upper (orange) and lower (blue) chain. The merge line (purple) connects both skeletons, the solid arcs are part of the merged skeleton.

For simplicity we assume  $\mathcal{P}$  to be  $x$ -monotone. We compute the straight skeleton of a single  $x$ -monotone chain incrementally by adding edge after edge, starting with the leftmost edge. During the construction we maintain a partial straight skeleton  $\mathcal{S}^*$ . For every edge  $e$  already added,  $\mathcal{S}^*$  maintains the left part of the boundary of the face of  $e$ . That is the left (orange) chain of the face  $f(e_i)$  in the example in Figure 9a. We show that every added edge only creates a single unbounded arc in  $\mathcal{S}^*$ . Additionally, the arcs created by adding a new edge only intersect unbounded arcs of  $\mathcal{S}^*$ . These two properties let us derive the linear bound on the construction of the straight skeleton of a single chain.

We show that all faces of  $\mathcal{P}$ , induced by the straight skeleton, are  $x$ -monotone. Additionally, the polygonal *merge line*, that is created by merging the skeletons of the two chains, is  $x$ -monotone. Therefore, we can walk along the boundaries of the respective faces while creating this merge line. We show that every arc is traversed at most once, which results in the linear time bound for the merge step. See an illustration of such a merge in Figure 9b. Together with the approach above we obtain an optimal linear time algorithm to compute the straight skeleton of an orthogonal monotone polygon.

### 2.3 WEIGHTED STRAIGHT SKELETONS AND A LINE ARRANGEMENT

In *Computing Positively Weighted Straight Skeletons of Simple Polygons based on Bisector Arrangement* [EH18a] (page 32), we study the weighted straight skeleton with support of a line arrangement. In recent straight skeleton research, there is much focus on the motorcycle graph, as it can be used to obtain the straight skeleton more easily [CV02, CV07, HH10, Hub12, HH12, MHH12, CMV16].

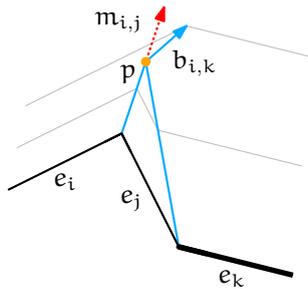


Figure 10: The weight of  $e_k$  is higher than the weight of both  $e_i$  and  $e_j$ ; the wavefront after  $p$ , on  $b_{i,k}$ , is reflex; the motorcycle  $m_{i,j}$ , induced by the input, stays on its initial direction.

A key insight in our work has been, that the motorcycle graph is not helpful in the weighted case: In the classical straight skeleton, the motorcycle graph can be utilized to track the reflex vertices of the wavefront propagation. Note, that only reflex wavefront vertices cause split events, which are difficult to predict. Every arc of the straight skeleton that is traced out by a reflex wavefront vertex lies on a motorcycle trace. This is no longer true for the weighted variant, where an event may cause a reflex wavefront vertex to change direction. See an illustration of such an event in Figure 10. Such a change in direction would require an update of the motorcycle graph. We can construct a scenario where

a linear number of updates are caused. Since a single update can result in the re-computation of the motorcycle graph, it is no longer a feasible aid.

Still, the idea to track the reflex wavefront vertices, i.e., their direction in the wavefront propagation is fundamental. Therefore, we use an arrangement of lines, that covers all reflex wavefront vertices. If such a wavefront vertex changes direction, we update the line arrangement.

To summarize our approach: We maintain the moving wavefront over time as well as the line arrangement embedded in the wavefront, illustrated in Figure 11 (blue and orange). Additionally, we keep track of the first and last intersection of a line in the arrangement in respect to the other arrangement lines. This enables us to identify events caused by reflex wavefront vertices more easily. Depending on how many intersections in the line arrangement we store the complexity of the algorithm changes and permits us to control a time/space trade-off.

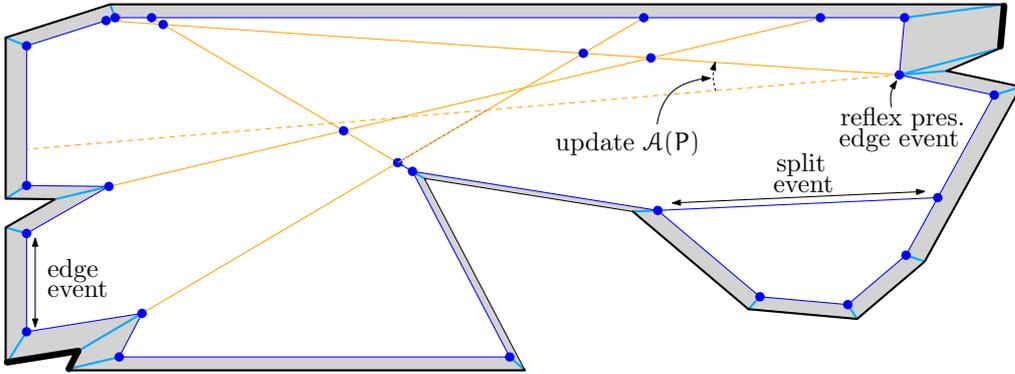


Figure 11: Extended wavefront example, wavefront (blue), line arrangement (orange), thicker input edges have higher weight, already traced area is highlighted in (gray).

Our algorithm is reflex sensitive and allows us to compute the weighted straight skeleton of a simple polygon in  $\mathcal{O}(n^2 + r^3/k + nr \log n)$  time and  $\mathcal{O}(n + kr)$  space, where  $r$  denotes the number of reflex vertices and  $0 < k \leq r$ . Our approach is a practical candidate for an implementation, as it does not require complex data structures. Additionally,  $k$  allows to control a time/space trade-off depending on specific used cases.

## 2.4 RECOGNIZING WEIGHTED STRAIGHT SKELETONS

In *Recognizing Geometric Trees as Positively Weighted Straight Skeletons and Reconstructing Their Input* [EHP19b] (page 34), we gain further insight into the structure of weighted straight skeletons. We study their geometric graph class the bisector graphs and their inducing input. We ask the question: Given a geometric graph  $G$  can we recognize  $G$  as a weighted straight skeleton and possibly

create an inducing input  $H$  and weight function  $\sigma$  for  $G$ ? Also, for which types of geometric graphs can we answer that question in the affirmative?

Our first contribution is an algorithm that constructs such an input  $H$  and  $\sigma$  in case  $G$  forms a geometric tree where the edges incident at every node induce a local convex partition. The basic idea is to apply a wavefront propagation in reverse. Therefore, we start at a node  $v$  of  $G$  and create an initial convex polygon where each corner resides on an edge incident at  $v$ . Then we propagate that polygon outwards and add edges whenever we sweep over a node of  $G$ . At the time this wavefront contains all nodes of  $G$  we have obtained a solution  $H$  and  $\sigma$ . Note, we also describe an approach if at most one node of  $G$  does not induce a local convex partition.

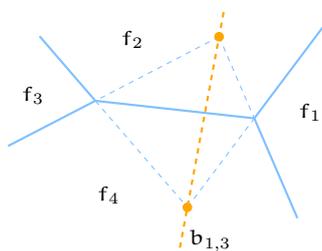


Figure 12: Construct the bisector  $b_{1,3}$  (orange) between faces,  $f_1$  and  $f_3$  with common neighbors.

Observe that  $H$  implicitly defines a quadratic number of bisectors, one for every edge pair. Our second contribution is a method to construct all bisectors that are not in  $G$  without the need to find an actual solution  $H$ . For our method to work we require all bisector-pairs to be non-parallel. Roughly, we construct a dual graph of  $G$  that helps in identifying which bisector lines we have to intersect in order to obtain a specific bisector that is not in  $G$ . Furthermore, we show that every solution  $H$  and  $\sigma$  that defines  $G$  also defines the same set of bisectors. In [Figure 12](#), we illustrate the construction of one bisector that is not in the original graph  $G$ .

## 2.5 WEIGHTED VORONOI DIAGRAMS IN THE MAXIMUM-NORM

In [Section 1.2](#), we describe the classical Voronoi diagram of point sites, which has a straight-line structure. Papadopoulou and Lee show in their work [[PL01](#)] how to compute the Voronoi diagram in the maximum norm of point and line-segment sites. The weighted Voronoi diagram of point sites in the  $L_2$ -norm consists of circular arcs [[AE84](#)], as illustrated in [Figure 13a](#).

In our work *Weighted Voronoi Diagrams in the  $L_\infty$ -Norm* [[EH19](#)] ([page 36](#)), we pick up this lead and investigate the diagram for points, line-segments, and rectangles. Preliminary work was presented at the Young Researchers Forum as part of CG Week in 2018 [[EH18b](#)]. Despite the weights and the various site shapes, this diagram is formed from straight-line segments only. In [Figure 13](#), we illustrate an example comparing the diagrams over the same set of points and weights in the  $L_2$ - and  $L_\infty$ -norm.

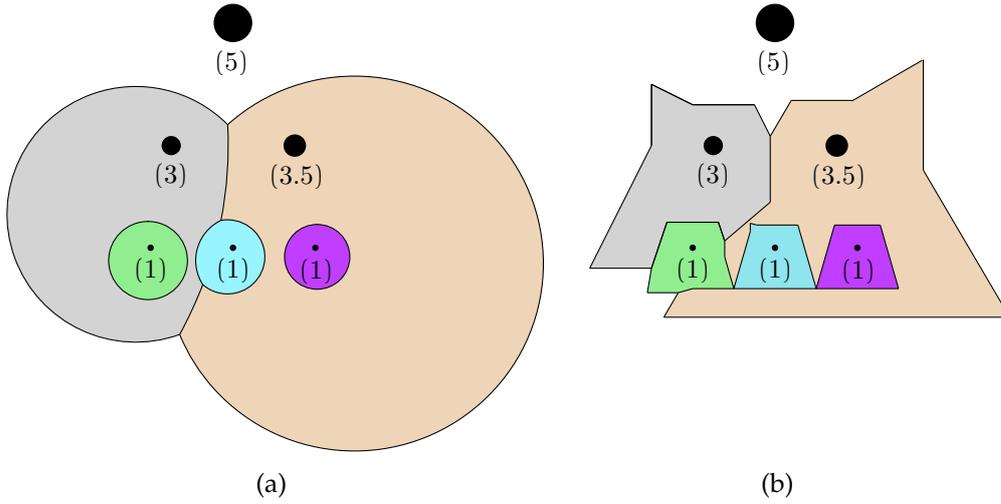


Figure 13: Multiplicatively weighted Voronoi diagram of six point sites, associated weight in brackets; (a) in  $L_2$ -norm; (b) in  $L_\infty$ -norm.

We construct the diagram applying an incremental construction approach. In each step we select a site  $s$  with maximal weight, from the sites not yet in the diagram. Then, we construct the Voronoi region of  $s$  and embed it into the diagram. We show that the site with minimal weight always defines a single connected region that contains the site itself. By selecting a site with maximal weight in each step, every added site has minimal weight among the sites already in the diagram.

We establish a worst-case combinatorial complexity bound of  $\mathcal{O}(n^2)$  if the sites include points and axis-aligned rectangles/line-segment. In this setting our construction approach takes  $\mathcal{O}(n^2 \log n)$  time and quadratic space. For general rectangles/line-segments we are able to derive a combinatorial complexity bound of  $\mathcal{O}(n^2 \alpha(n))$ , where  $\alpha(n)$  denotes the inverse Ackermann function.

## BIBLIOGRAPHY

- [AA96] Oswin Aichholzer and Franz Aurenhammer: *Straight Skeletons for General Polygonal Figures in the Plane*. In *Proceedings of the 2nd Annual International Conference on Computing and Combinatorics (COCOON 1996)*. Lecture Notes in Computer Science, volume 1090, Springer-Verlag, June 1996, pages 117–126.  
DOI: [10.1007/3-540-61332-3\\_144](https://doi.org/10.1007/3-540-61332-3_144)
- [AAAG95] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner: *A Novel Type of Skeleton for Polygons*. In *Journal of Universal Computer Science*, volume 1 (12), 1995, pages 752–761.  
DOI: [10.1007/978-3-642-80350-5\\_65](https://doi.org/10.1007/978-3-642-80350-5_65)
- [AE84] Franz Aurenhammer and Herbert Edelsbrunner: *An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane*. In *Pattern Recognition*, volume 17 (2), 1984, pages 251–257.  
DOI: [10.1016/0031-3203\(84\)90064-5](https://doi.org/10.1016/0031-3203(84)90064-5)
- [Bie+15a] Therese Biedl, Martin Held, Stefan Huber, Dominik Kaaser, and Peter Palfrader: *A Simple Algorithm for Computing Positively Weighted Straight Skeletons of Monotone Polygons*. In *Information Processing Letters*, volume 115 (2), February 2015, pages 243–247.  
DOI: [10.1016/j.ipl.2014.09.021](https://doi.org/10.1016/j.ipl.2014.09.021)
- [Bie+15b] Therese Biedl, Martin Held, Stefan Huber, Dominik Kaaser, and Peter Palfrader: *Weighted Straight Skeletons in the Plane*. In *Computational Geometry: Theory and Applications*, volume 48 (2), February 2015, pages 120–133.  
DOI: [10.1016/j.comgeo.2014.08.006](https://doi.org/10.1016/j.comgeo.2014.08.006)
- [Blu67] Harry Blum: *A Transformation for Extracting New Descriptors of Shape*. In *Models for the Perception of Speech and Visual Form*. MIT Press, 1967, pages 362–380.
- [BTKA10] Martin Bock, Amit K. Tyagi, Jan-Ulrich Kreft, and Wolfgang Alt: *Generalized Voronoi Tessellation as a Model of Two-dimensional Cell Tissue Dynamics*. In *Bulletin of Mathematical Biology*, volume 72 (7), October 2010, pages 1696–1731.  
DOI: [10.1007/s11538-009-9498-3](https://doi.org/10.1007/s11538-009-9498-3)
- [Cha93] Bernard Chazelle: *Cutting hyperplanes for divide-and-conquer*. In *Discrete & Computational Geometry*, volume 9 (2), February 1993, pages 145–158.  
DOI: [10.1007/BF02189314](https://doi.org/10.1007/BF02189314)
- [CMV16] Siu-Wing Cheng, Liam Mencil, and Antoine Vigneron: *A Faster Algorithm for Computing Straight Skeletons*. In *ACM Transactions on Algorithms*, volume 12 (3), April 2016, pages 44:1–44:21.  
DOI: [10.1145/2898961](https://doi.org/10.1145/2898961)

- [CV02] Siu-Wing Cheng and Antoine Vigneron: *Motorcycle Graphs and Straight Skeletons*. In *Proceedings of the 13th Symposium on Discrete Algorithms (SODA 2002)*, 2002, pages 156–165.
- [CV07] Siu-Wing Cheng and Antoine Vigneron: *Motorcycle Graphs and Straight Skeletons*. In *Algorithmica*, volume 47(2), February 2007, pages 159–182.  
DOI: [10.1007/s00453-006-1229-7](https://doi.org/10.1007/s00453-006-1229-7)
- [CGAL] *Computational Geometry Algorithms Library*.  
URL: <http://www.cgal.org/>
- [DK97] Frank Dehne and Rolf Klein: “The big sweep”: *On the power of the wavefront approach to Voronoi diagrams*. In *Algorithmica*, volume 17(1), January 1997, pages 19–32.  
DOI: [10.1007/BF02523236](https://doi.org/10.1007/BF02523236)
- [DDL99] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw: *Folding and One Straight Cut Suffice*. In *Proceedings of the 10th Symposium on Discrete Algorithms (SODA 1999)*, January 1999, pages 891–892.
- [Des44] René Descartes: *Renati Des-Cartes Principia philosophiae*. Amstelodami, apud Ludovicum Elzevirium, 1644.  
URL: <https://lccn.loc.gov/46028327>
- [Dir50] Johann P. G. Lejeune Dirichlet: *Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen*. In *Journal für die reine und angewandte Mathematik*, volume 40, 1850, pages 209–227.  
URL: <http://eudml.org/doc/147457>
- [EH18a] Günther Eder and Martin Held: *Computing Positively Weighted Straight Skeletons of Simple Polygons based on Bisector Arrangement*. In *Information Processing Letters*, volume 132, 2018, pages 28–32.  
ISSN: 0020-0190.  
DOI: [10.1016/j.ipl.2017.12.001](https://doi.org/10.1016/j.ipl.2017.12.001)
- [EH18b] Günther Eder and Martin Held: *Weighted Voronoi Diagrams in the  $L_\infty$ -Norm*. In *Proceedings of the 7th Young Researchers Forum (CG Week 2018)*, June 2018.
- [EH19] Günther Eder and Martin Held: *Weighted Voronoi Diagrams in the  $L_\infty$ -Norm*. Submitted to the *International Journal of Computational Geometry*. January 2019.
- [EHP16] Günther Eder, Martin Held, and Peter Palfrader: *Bisector Graphs for Min-/Max-Volume Roofs over Simple Polygons*. In *Proceedings of the 32st European Workshop on Computational Geometry (EuroCG 2016)*, March 2016.

- [EHP18] Günther Eder, Martin Held, and Peter Palfrader: *Min-/Max-Volume Roofs Induced by Bisector Graphs of Polygonal Footprints of Buildings*. In *International Journal of Computational Geometry*, volume 28 (04), 2018, pages 309–340.  
DOI: [10.1142/S0218195918500097](https://doi.org/10.1142/S0218195918500097)
- [EHP19a] Günther Eder, Martin Held, and Peter Palfrader: *Computing the Straight Skeleton of an Orthogonal Monotone Polygon in Linear Time*. In *Proceedings of the 35th European Workshop on Computational Geometry (EuroCG 2019)*, March 2019.
- [EHP19b] Günther Eder, Martin Held, and Peter Palfrader: *Recognizing Geometric Trees as Positively Weighted Straight Skeletons and Reconstructing Their Input*. Submitted to the *International Journal of Computational Geometry*. April 2019.
- [EE99] David Eppstein and Jeff Erickson: *Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions*. In *Discrete & Computational Geometry*, volume 22 (4), 1999, pages 569–592.  
DOI: [10.1145/276884.276891](https://doi.org/10.1145/276884.276891)
- [For86] Steven J. Fortune: *A Sweepline Algorithm for Voronoi Diagrams*. In *Proceedings of the Second Annual Symposium on Computational Geometry*. SCG '86. ACM, 1986, pages 313–322.  
ISBN: 0-89791-194-6.  
DOI: [10.1145/10515.10549](https://doi.org/10.1145/10515.10549)
- [GMAM06] Santiago Garrido, Luis Moreno, Mohamed Abderrahim, and Fernando Martin: *Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching*. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, pages 2376–2381.  
DOI: [10.1109/IROS.2006.282649](https://doi.org/10.1109/IROS.2006.282649)
- [HSo8] Jan-Henrik Haunert and Monika Sester: *Area Collapse and Road Centerlines based on Straight Skeletons*. In *GeoInformatica*, volume 12 (2), June 2008, pages 169–191.  
DOI: [10.1007/s10707-007-0028-x](https://doi.org/10.1007/s10707-007-0028-x)
- [Hub12] Stefan Huber: *Computing Straight Skeletons and Motorcycle Graphs: Theory and Practice*. Shaker Verlag, April 2012.  
ISBN: 978-3-8440-0938-5.
- [HH10] Stefan Huber and Martin Held: *Computing Straight Skeletons of Planar Straight-Line Graphs Based on Motorcycle Graphs*. In *Proceedings of the 22th Canadian Conference on Computational Geometry (CCCG 2010)*, August 2010, pages 187–190.

- [HH12] Stefan Huber and Martin Held: *A Fast Straight-Skeleton Algorithm Based On Generalized Motorcycle Graphs*. In *International Journal of Computational Geometry*, volume 22 (5), October 2012, pages 471–498.  
DOI: [10.1142/S0218195912500124](https://doi.org/10.1142/S0218195912500124)
- [Lat91] Jean-Claude Latombe: *Robot Motion Planning*. Springer, 1991.  
DOI: [10.1007/978-1-4615-4022-9](https://doi.org/10.1007/978-1-4615-4022-9)
- [LDo3] Robert G. Laycock and Andrew M. Day: *Automatically Generating Large Urban Environments based on the Footprint Data of Buildings*. In *Proceedings of the 8th ACM Symposium on Solid Modeling and Applications (SM 2003)*, June 2003, pages 346–351.  
DOI: [10.1145/781606.781663](https://doi.org/10.1145/781606.781663)
- [MHH12] Willi Mann, Martin Held, and Stefan Huber: *Computing Motorcycle Graphs Based on Kinetic Triangulations*. In *Proceedings of the 24th Canadian Conference on Computational Geometry (CCCG 2012)*, August 2012, pages 187–192.
- [OBSCoo] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung N. Chiu: *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. John Wiley & Sons, Inc, 2000.  
ISBN: 978-0471986355.  
DOI: [10.1002/9780470317013](https://doi.org/10.1002/9780470317013)
- [PH15] Peter Palfrader and Martin Held: *Computing Mitered Offset Curves Based on Straight Skeletons*. In *Computer-Aided Design and Applications*, volume 12 (4), February 2015, pages 414–424.  
DOI: [10.1080/16864360.2014.997637](https://doi.org/10.1080/16864360.2014.997637)
- [PHH12] Peter Palfrader, Martin Held, and Stefan Huber: *On Computing Straight Skeletons by Means of Kinetic Triangulations*. In *Proceedings of the 20th Annual European Symposium on Algorithms (ESA 2012)*. Lecture Notes in Computer Science, volume 7501, Springer-Verlag, September 2012, pages 766–777.  
DOI: [10.1007/978-3-642-33090-2\\_66](https://doi.org/10.1007/978-3-642-33090-2_66)
- [PLo1] Evanthia Papadopoulou and D.T. Lee: *The  $L_\infty$  Voronoi Diagram of Segments and VLSI Applications*. In *International Journal of Computational Geometry*, volume 11 (05), 2001, pages 503–528.  
DOI: [10.1142/S0218195901000626](https://doi.org/10.1142/S0218195901000626)
- [Pes77] Gustav A. V. Peschka: *Kotirte Ebenen (Kotirte Projektionen) und deren Anwendung : Vorträge : zum Gebrauche für Ingenieure, für höhere Lehranstalten und zum Selbststudium : mit 47 lithogr. Tafeln*. Technische Informationsbibliothek (TIB), 1877.  
DOI: [10.14463/GBV:865177619](https://doi.org/10.14463/GBV:865177619)
- [RO] *Roofer - Computing Min/Max-Volume Roofs*.  
URL: <https://github.com/guenthereder/roofer>

- [Sán+16] Daniel Sánchez-Gutiérrez, Melda Tozluoglu, Joseph D. Barry, Alberto Pascual, Yanlan Mao, and Luis M. Escudero: *Fundamental physical cellular constraints drive self-organization of tissues*. In *The EMBO Journal*, volume 35 (1), 2016, pages 77–88.  
DOI: [10.15252/emj.201592374](https://doi.org/10.15252/emj.201592374)
- [SH75] Michael I. Shamos and Dan Hoey: *Closest-point Problems*. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science (FOCS 1975)*. IEEE Computer Society, 1975, pages 151–162.  
DOI: [10.1109/SFCS.1975.8](https://doi.org/10.1109/SFCS.1975.8)
- [SK18] Kenichi Sugihara and Youry Khmelevsky: *Roof report from automatically generated 3D building models by straight skeleton computation*. In *2018 Annual IEEE International Systems Conference (SysCon)*, April 2018, pages 1–8.  
DOI: [10.1109/SYSCON.2018.8369554](https://doi.org/10.1109/SYSCON.2018.8369554)
- [TS12] Akiyasu Tomoeda and Kokichi Sugihara: *Computational Creation of a New Illusionary Solid Sign*. In *Proceedings of the 9th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2012)*. IEEE Computer Society, June 2012, pages 144–147.  
DOI: [10.1109/ISVD.2012.26](https://doi.org/10.1109/ISVD.2012.26)
- [Voro8] Georgy F. Voronoi: *Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites*. In *Journal für die reine und angewandte Mathematik*, volume 133, 1908, pages 97–178.  
URL: <http://eudml.org/doc/149276>



# Part II

## PUBLICATIONS

Note:

The abridged version of this work does not contain full copies of the papers that comprise this cumulative thesis. Instead, only references are provided.

MIN-/MAX-VOLUME ROOFS INDUCED BY BISECTOR  
GRAPHS OF POLYGONAL FOOTPRINTS OF BUILDINGS

---

Günther Eder, Martin Held, and Peter Palfrader [EHP18].

---

[EHP18]: Günther Eder, Martin Held, and Peter Palfrader: *Min-/Max-Volume Roofs Induced by Bisector Graphs of Polygonal Footprints of Buildings*. In *International Journal of Computational Geometry*, volume 28 (04), 2018, pages 309–340.  
DOI: [10.1142/S0218195918500097](https://doi.org/10.1142/S0218195918500097)

[minmaxroofs.pdf]

COMPUTING THE STRAIGHT SKELETON OF AN  
ORTHOGONAL MONOTONE POLYGON IN LINEAR TIME

---

Günther Eder, Martin Held, and Peter Palfrader [EHP19a].

---

[EHP19a]: Günther Eder, Martin Held, and Peter Palfrader: *Computing the Straight Skeleton of an Orthogonal Monotone Polygon in Linear Time*. In *Proceedings of the 35th European Workshop on Computational Geometry (EuroCG 2019)*, March 2019

[ompoly-sk-lin.pdf]

COMPUTING POSITIVELY WEIGHTED STRAIGHT  
SKELETONS OF SIMPLE POLYGONS BASED ON A  
BISECTOR ARRANGEMENT

---

Günther Eder and Martin Held [[EH18a](#)].

---

[[EH18a](#)]: Günther Eder and Martin Held: *Computing Positively Weighted Straight Skeletons of Simple Polygons based on Bisector Arrangement*. In *Information Processing Letters*, volume 132, 2018, pages 28–32. ISSN: 0020-0190.  
DOI: [10.1016/j.ipl.2017.12.001](https://doi.org/10.1016/j.ipl.2017.12.001)

[wss-arrangement.pdf]

RECOGNIZING GEOMETRIC TREES AS POSITIVELY  
WEIGHTED STRAIGHT SKELETONS AND  
RECONSTRUCTING THEIR INPUT

---

Günther Eder, Martin Held, and Peter Palfrader [EHP19b].

---

[EHP19b]: Günther Eder, Martin Held, and Peter Palfrader: *Recognizing Geometric Trees as Positively Weighted Straight Skeletons and Reconstructing Their Input*. Submitted to the *International Journal of Computational Geometry*. April 2019

[recog-wss.pdf]

MULTIPLICATIVELY WEIGHTED VORONOI DIAGRAMS IN  
THE MAXIMUM NORM

---

Günther Eder and Martin Held [EH19].

---

[EH19]: Günther Eder and Martin Held: *Weighted Voronoi Diagrams in the  $L_\infty$ -Norm*. Submitted to the International Journal of Computational Geometry. January 2019

[ijcga-mwvd.pdf]