# Weighted Skeletal Structures For Computing Variable-Radius Offsets

Martin Held[1], Stefan de Lorenzo[2]

[1]University of Salzburg, held@cs.sbg.ac.at
[2]University of Salzburg, slorenzo@cs.sbg.ac.at

Corresponding author: Stefan de Lorenzo, slorenzo@cs.sbg.ac.at

**Abstract.** Held et al. [CAD&A 2016] introduced generalized weighted Voronoi diagrams as a tool to construct variable-radius offsets. We revisit this structure and provide two algorithms to compute it. We also introduce variable-radius skeletons of polygons as a closely related structure which inherits most properties of generalized weighted Voronoi diagrams except that its skeletal regions are always connected. Experimental results obtained by our implementation of variable-radius skeletons are discussed. In addition to variable-radius offsets we demonstrate how this structure can be used to generate intricate roofs for polygonal footprints of buildings in an automated way.

**Keywords:** Voronoi diagram, multiplicative weight, wavefront propagation, variable-radius skeleton, variable-radius offsets
**DOI:** https://doi.org/10.14733/cadaps.2021.875-889

## 1 Introduction

Offsetting is an essential task in many industrial applications. In conventional *constant-radius offsetting* all parts of the input set expand or shrink uniformly at the same speed. A natural extension of this idea is to allow parts to expand or shrink in a non-uniform manner. Among other fields, such so-called *variable-radius offsets* find applications in brush-stroke modeling and the generation of ornamental seams. See Figure 1 for sample constant-radius and variable-radius offsets of a set of straight-line segments that are arranged in a star-like fashion. In the sequel we present skeletal structures that support the computation of variable-radius offsets.

Consider a set $S^*$ of $n$ points in the plane. The *Voronoi diagram* $\mathcal{VD}(S^*)$ of $S^*$ partitions the plane into interior-disjoint, convex areas — so-called Voronoi regions — such that the Voronoi region $\mathcal{VR}(s, S^*)$ of $s \in S^*$ contains all points of the plane that are closer to $s$ under the Euclidean distance metric than to any other point of $S^*$; see Figure 2a. It is well-known that $\mathcal{VD}(S^*)$ has $\mathcal{O}(n)$ nodes and (straight-line) edges. In the well-known prairie-fire analogy, $\mathcal{VD}(S^*)$ is given by those points of the plane where fire waves meet

- if fires are ignited simultaneously at all points of $S^*$, and
- if all fires spread uniformly at the same speed.

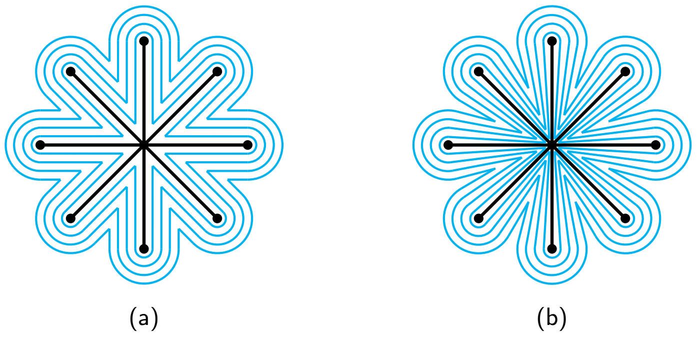


**Figure 1**: A family of constant-radius and variable-radius offsets (in blue) of straight-line segments (in black).

These fire waves form instances of (constant-radius) offset curves of the points of $S^*$ for specific offset distances; see Figure 2b. In the field of computational geometry, offset curves are known as *wavefronts*, and a simulation of the wavefronts for steadily increasing offset distances is called *wavefront propagation*.

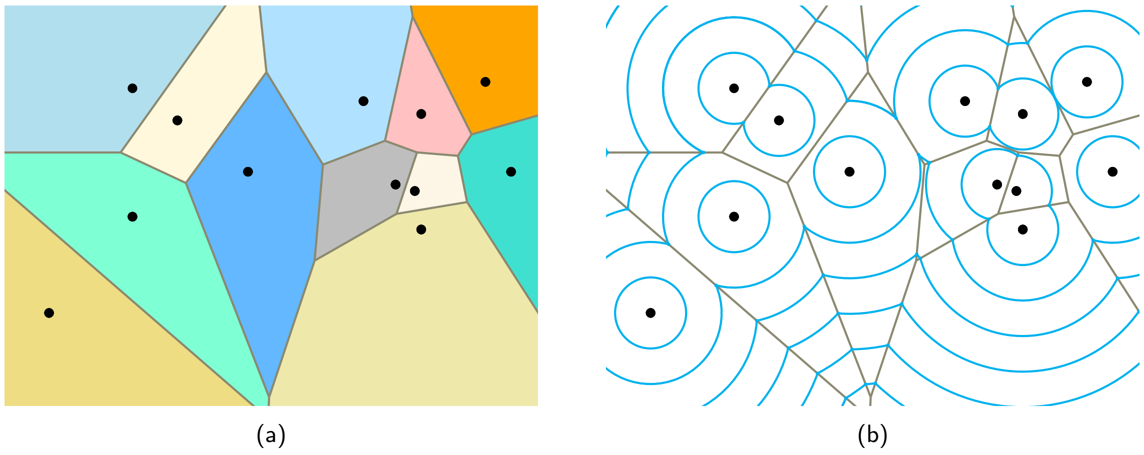


**Figure 2**: (a) Voronoi diagram and (b) a family of wavefronts for a set of points. Each Voronoi region is indicated by a colored area.

In more formal terms,

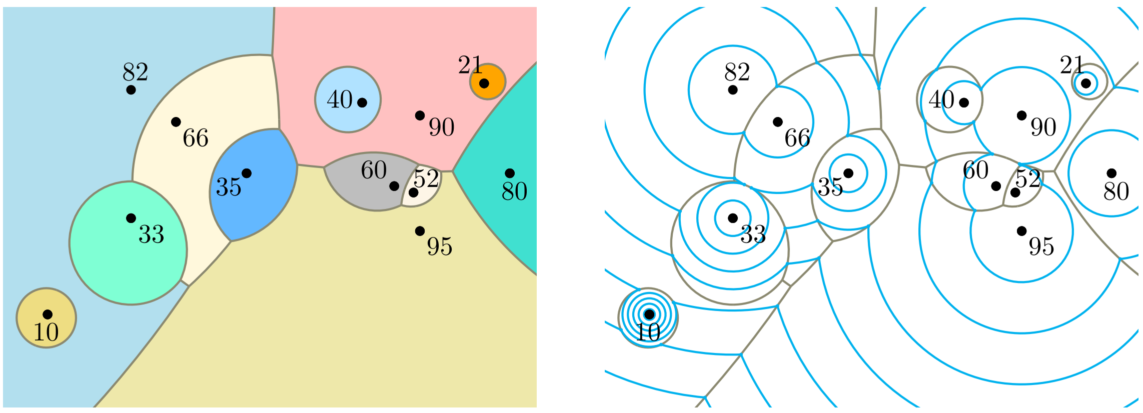
$$\mathcal{VD}(S^*) := \bigcup_{s \in S^*} \partial \mathcal{VR}(s, S^*),$$

where

$$\mathcal{VR}(s, S^*) := \{p \in \mathbb{R}^2 : d(p,s) \le d(p, S^*)\},$$

and $\partial\mathcal{VR}(s, S^*)$ denotes the boundary of $\mathcal{VR}(s, S^*)$. Lots of generalizations of the standard Voronoi diagram have been studied. See, e,g., generalizations in terms of dimensionality [7], the types of permissible input objects [10], and distance metric [14].

Another way to generalize Voronoi diagrams is to assign multiplicative positive weights to the points of $S^*$ and to let them influence the expansion speeds of the wavefronts. This yields the *multiplicatively weighted Voronoi diagram* of $S^*$; see Figure 3a. More formally, the weighted distance $d_w(p, s)$ between a point $p \in \mathbb{R}^2$



(a) (b)

**Figure 3**: (a) The multiplicatively weighted Voronoi diagram and (b) corresponding family of wavefronts for a set of weighted points. The multiplicative weights are written next to the points; their positions are identical to the positions shown in Figure 2a.

and a weighted point $s \in S^*$ is defined as

$$d_w(p, s) := \frac{d(p, s)}{w(s)},$$

where $d(p, s)$ denotes the standard Euclidean distance between $p$ and $s$, and $w(s)$ is the positive weight of $s$. Hence, the larger the weight of a point the quicker its fire wavefront spreads.

Held et al. [12] generalize this concept and introduce the *generalized weighted Voronoi diagram* (GWVD) of a set $S$ of weighted points and variably-weighted straight-line segments: They assign weights to the end-points $a$ and $b$ of an input segment $\overline{ab}$ and then obtain the weight of a point $q$ on $\overline{ab}$ by a linear interpolation of the weights of $a$ and $b$. Then

$$d_w(p, \overline{ab}) := \min_{q \in \overline{ab}} d_w(p, q),$$

and the corresponding GWVD $\mathcal{VD}_w(S)$ is defined accordingly. See Figure 4 for a sample GWVD and corresponding wavefronts, i.e., variable-radius offsets. We note that even simple differences in the weights may have a significant impact. For instance, the variable-radius offsets shown in Figure 1b were achieved by assigning all other end-points twice the weight of the center point.

## 2 Our Contribution

We start with examining the generalized weighted Voronoi diagram (GWVD) in more detail: We define an explicit distance function between an arbitrary point in the plane and a variably-weighted straight-line segment. Additionally, two different strategies to compute the GWVD are outlined.

Since the individual Voronoi regions of a GWVD may be disconnected, we introduce a closely related structure whose regions stay connected. We call this structure a variable-radius skeleton (VRS), present an algorithm for computing a VRS inside of a polygon, and discuss results obtained by our prototype implementation.

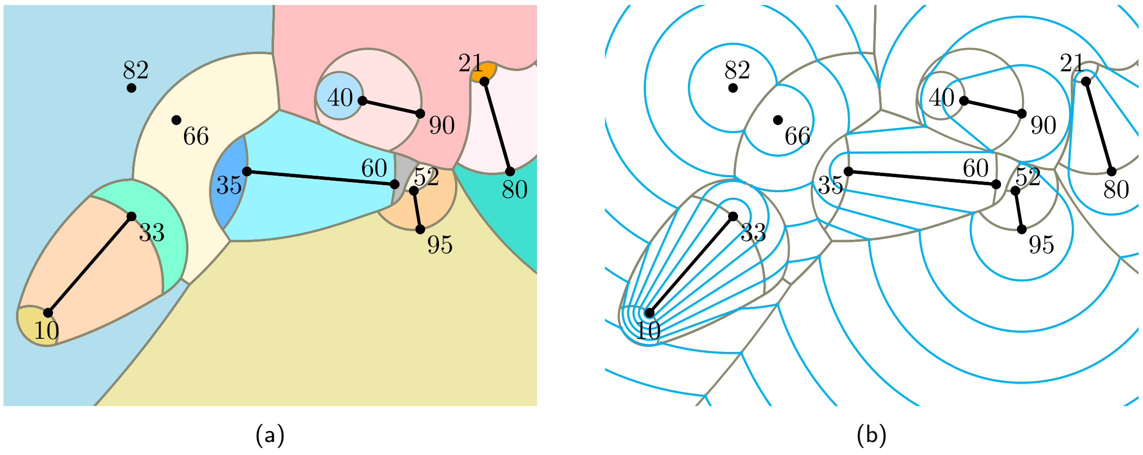


**Figure 4**: (a) Generalized weighted Voronoi diagram of a set of weighted points and straight-line segments (highlighted in black) and (b) family of corresponding wavefronts. All points have the same positions and weights as in Figure 3a.

## 3 Preliminaries

Let $S$ be a set of weighted points and variably-weighted straight-line segments. All multiplicative weights are required to be positive. (Biedl et al. [5] show that a weighted skeletal structure may lose virtually all important properties of its unweighted sibling if negative weights are allowed.) No input point is allowed to lie on a line segment, and no pair of line segments may share a point except for a common end-point. All input segments and input points are called *sites*. For the sake of descriptional simplicity, it is assumed that no point in $\mathbb{R}^2$ has the same weighted distance to more than three sites of $S$.

Every input site $s \in S$ is associated with a so-called *offset circle* $c(s,t)$ which includes all points in $\mathbb{R}^2$ that are at weighted distance $t$ to $s$. We find it convenient to regard $c(s,t)$ as a function of either time or distance since at time $t$ every point on $c(s,t)$ is at Euclidean distance $t \cdot w(s)$ from $s$, i.e., at weighted distance $t$. As discussed in [12], the offset circle $c(\overline{ab},t)$ of a straight-line segment $\overline{ab}$ is formed by two circular arcs, which are induced by its end-points $a$ and $b$, and two straight-line segments; see Figure 5c. (Of course, the offset circle of a variably-weighted straight-line segment is no genuine circle but we prefer to use the same term for the offsets of both points and straight-line segments.) A similar result holds for the offset circle of a circular arc if identical weights are assigned to its end-points.

The wavefront $\mathcal{W}(S,t)$ emanated by $S$ at time $t \geq 0$ is the set of all points $p$ of the plane whose minimal weighted distance from $S$ equals $t$. More formally,

$$\mathcal{W}(S,t) := \left\{ p \in \mathbb{R}^2 : \min_{s \in S} d_w(p,s) = t \right\}.$$

Hence, for $t = 0$ the wavefront $\mathcal{W}(S,t)$ equals $S$. All wavefronts consist of portions of offset circles and, thus, consist only of straight-line segments and circular arcs. Every such *wavefront edge* is associated with its corresponding input site. A common end-point of two adjacent wavefront edges is called a *wavefront vertex*. Similar to standard Voronoi diagrams and straight skeletons [16], the wavefront vertices will trace out the edges of our skeletal structures. That is, they move along the bisectors of pairs of input sites (relative to the weighted distance).

Consider two sites $s_1, s_2 \in S$ and assume that their offset circles intersect. If $s_1$ and $s_2$ both are weighted points then their offset circles intersect in exactly one pair of points, which we call *moving intersections*. (These

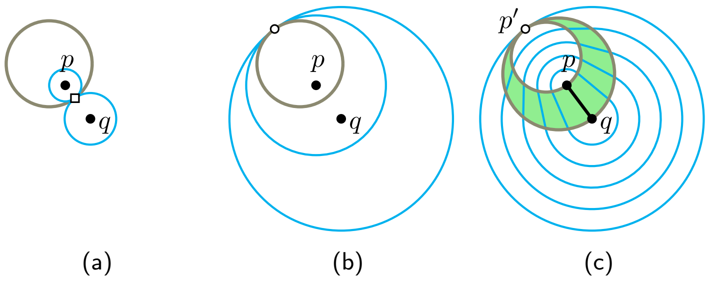


**Figure 5**: In (a) and (b) the collision and domination points, respectively, of two offset circles that are emanated by two weighted points $p$ and $q$ are displayed for $w(p) < w(q)$. In (c) $p$ and $q$ are the end-points of a variably-weighted straight-line segment $\overline{pq}$. The singular point $p'$ of $\overline{pq}$ is indicated by a small black circle. The circular boundaries of the corresponding areas of influence are shown in dark green, and $\mathcal{AOI}(\widetilde{pq})$ is shaded in light green.

points may coincide, though.) Otherwise, the offset circles may define up to two pairs of moving intersections. Every such a moving intersection traces out a part of the bisector $b(s_1, s_2)$ of $s_1$ and $s_2$. These traces are non-overlapping (except possibly for the respective start- or end-points) and their union equals $b(s_1, s_2)$. We say that $c(s_1, t)$ and $c(s_2, t)$ *collide* at time $t$ if a pair of moving intersections appears for the first time at time $t$. If a pair of moving intersections disappears at time $t$ then $c(s_1, t)$ *dominates* $c(s_2, t)$ at $t$; see Figure 5.

To avoid two-dimensional bisectors between two variably-weighted straight-line segments that share a common end-point, we adapt Held's concept of *areas of influence* [11]: Every variably-weighted straight-line segment $\overline{pq}$ induces a subdivision of the plane into three areas which we denote by $\mathcal{AOI}(p)$, $\mathcal{AOI}(q)$, and $\mathcal{AOI}(\widetilde{pq})$, respectively, where $\widetilde{pq}$ is the open straight-line segment between $p$ and $q$, i.e., $\overline{pq}$ without its end-points. Each area of influence includes the points of $\mathbb{R}^2$ that are closest to the corresponding portion of $\overline{pq}$. If $w(p) < w(q)$ then the areas of influence are bounded by two circles that touch in a single point $p'$ which we refer to as *singular point*; see Figure 5c. Note that $p'$ coincides with the point at which $q$ starts to dominate $p$. If $w(p) = w(q)$ then the areas of influence are bounded by two lines that are perpendicular to $\overline{pq}$ and run through $p$ and $q$.

## 4 Generalized Weighted Voronoi Diagrams

Algorithmic paradigms such as plane sweep [6] and incremental construction [15], which have been used extensively for computing Voronoi diagrams of unweighted as well as constantly weighted input sites, seem inapplicable for computing GWVDs. This is due to two main reasons. First of all, the individual Voronoi regions of a GWVD are (in general) not connected; see also Figure 6. Furthermore, it is (in general) not possible to establish an insertion order $(s_1, s_2, \dots, s_n)$ of the elements of $S$ such that the Voronoi region $\mathcal{VR}_w(s_i, S_i)$ of the $i$-th site $s_i$ in the Voronoi diagram $\mathcal{VD}_w(S_i)$ of the $i$-th prefix set $S_i := (s_1, s_2, \dots, s_i)$ stays connected for all $i \in \{1, 2, \dots, n\}$. Therefore, we focus on two avenues for computing GWVDs that are more resilient to the inherent properties of this structure.

Edelsbrunner and Seidel [7] establish the connection between Voronoi diagrams in $\mathbb{R}^d$ and lower envelopes in $\mathbb{R}^{d+1}$. Agarwal et al. [3] present an algorithm for computing the lower envelope of a set of $n$ algebraic bivariate functions in $\mathcal{O}(n^{2+\varepsilon})$ time[1], for any $\varepsilon > 0$. In order to employ this result we need to define an algebraic bivariate function that models the distance of a point to a specific input site, i.e., to a weighted point or straight-line segment. If $0 < w(a) = w(b)$ then the distance from a point $p \in \mathbb{R}^2$ to the straight-line

[1]A term of the form $k + \varepsilon$ means that there is some additive positive constant $\varepsilon$ that needs to be added to $k$ that may be regarded as arbitrarily small but it will never equal zero.

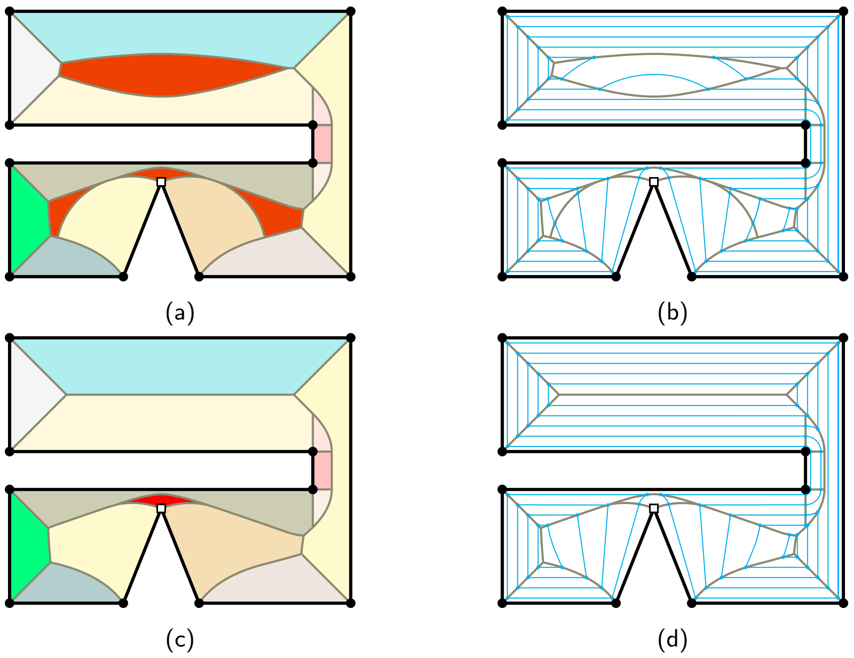


**Figure 6**: (a) GWVD and (b) corresponding wavefronts inside a polygon $P$ where the node indicated by a square has three times the weight of all other nodes. Note that the Voronoi region of the "square" node consists of four connected components shown in red. (c) VRS and (d) corresponding wavefronts.

segment $\overline{ab}$ is simply given by $d(p, \overline{ab})/w(a)$. Otherwise, assume that $0 < w(a) < w(b)$. For the sake of mathematical simplicity we further assume that $\overline{ab}$ lies on the positive $x$-axis such that the singular point $a'$ coincides with the coordinate origin. The closest weighted point $\psi_p\left(\overline{ab}\right)$ of $p$ on $\overline{ab}$ is given as

$$\psi_p\left(\overline{ab}\right) := \begin{cases} a & p \in \mathcal{AOI}(a), \\ b & p \in \mathcal{AOI}(b), \\ \left(\frac{x^2+y^2}{x}, 0\right) & \text{otherwise.} \end{cases}$$

This gives us

$$d_w(p, \overline{ab}) := d_w(p, \psi_p\left(\overline{ab}\right))$$

as the weighted distance $d_w(p, \overline{ab})$ between $p$ and $\overline{ab}$. Of course, every weighted straight-line segment can be aligned easily with the $x$-axis such that its singular point coincides with the origin. Thus, every input site is associated with an algebraic bivariate distance function. Held et al. [12] argue that the graph of such a distance function is a sub-surface of a right conoid.

Lemma 4.1 summarizes the fact that the general-purpose strategy of Agarwal et al. [3] can be utilized to compute GWVDs. Lemma 4.2 and Lemma 4.3 establish upper and lower bounds, respectively, on the combinatorial complexity of a GWVD in the worst case.

**Lemma 4.1.** The GWVD $\mathcal{VD}_w(S)$ of a set $S$ of $n$ weighted points and variably-weighted straight-line segments as input sites can be computed in $\mathcal{O}(n^{2+\varepsilon})$ time, for any $\varepsilon > 0$.

**Lemma 4.2.** The GWVD $\mathcal{VD}_w(S)$ of a set $S$ of $n$ input sites has a combinatorial complexity of $\mathcal{O}(n^{2+\varepsilon})$ in the worst case, for any $\varepsilon > 0$.

*Proof.* This bound is derived from the combinatorial complexity of the corresponding lower envelope. □

**Lemma 4.3.** The GWVD $\mathcal{VD}_w(S)$ of a set $S$ of $n$ input sites has a combinatorial complexity of $\Omega(n^2)$ in the worst case.

*Proof.* The multiplicatively weighted Voronoi diagram of points has a combinatorial complexity of $\Omega(n^2)$ in the worst case [4]. □

Even though lifting a problem to higher dimensions can be enlightening from a theoretical point of view, experience tells us that it often complicates matters when it comes to developing a practical implementation. And, indeed, the proof-of-concept implementation by Held et al. [12] that is based on this approach cannot go beyond tiny input sets. (And this limitation would hardly go away if their code were tuned for speed.)

Hence, we discuss an alternative wavefront-based strategy that operates entirely in the plane. Recall that the wavefront $\mathcal{W}(S,t)$ equals $S$ for $t = 0$. Starting at $t = 0$, the expansion of the wavefront $\mathcal{W}(S,t)$ is simulated by continuously increasing the time (or weighted distance) $t$. We do already know that $\mathcal{W}(S,t)$ consists of (possibly multiple) closed curvilinear chains, for all $t \geq 0$. If all segments of $S$ are disjoint then, for a sufficiently small time $t_0$, the wavefront $\mathcal{W}(S,t_0)$ consists of exactly two straight-line segments per input segment and of up to one circular arc per input point or end-point of a segment.

It is obvious that the vertices of $\mathcal{W}(S,t)$ change their positions as $t$ is increased. During the wavefront propagation process, we keep track of these moving intersections. However, a change of the locations of the wavefront vertices is not the only change that will happen. Rather, new wavefront edges may appear, old wavefront edges may disappear, and the wavefront may split into two or more connected wavefront components. These combinatorial changes are witnessed by the following three types of events.

**Definition 4.1** (Collision event)**.** A *collision event* occurs whenever a new pair of moving intersections appears.

**Definition 4.2** (Domination event)**.** A *domination event* occurs whenever a pair of moving intersections disappears.

**Definition 4.3** (Arc event)**.** An *arc event* occurs whenever two moving intersections coincide along an offset circle in such a way that this constellation can be neither categorized as a collision nor as a domination event.

An arc event may cause a wavefront edge to shrink to zero length and, thus, to disappear. As in the case of standard Voronoi diagrams, a new Voronoi node has been discovered whenever an old wavefront edge disappears or a new wavefront edge appears.

Every offset circle $c(s,t)$ holds the set of moving intersections at time $t$ sorted in counter-clockwise angular order around the respective site in a self-balancing binary search-tree. In particular, it stores those portions of $c(s,t)$ which overlap with $\mathcal{W}(S,t)$. Furthermore, every moving intersection holds a flag that indicates whether it is a wavefront vertex. Initially, all collision and domination events are computed and inserted into a priority queue $\mathcal{Q}$. All offset circles are initially empty. Afterwards, all events are successively retrieved from $\mathcal{Q}$.

- If a collision or domination event occurs at time $t_\sigma$ then a pair of moving intersections is inserted or removed, respectively, from their corresponding offset circles. If a collision event takes place on $\mathcal{W}(S,t_\sigma)$ then the newly created moving intersections are marked to be wavefront vertices.
- If an arc event takes place then three offset circles coincide at a single point. A Voronoi node has been discovered if at least one of the respective moving intersections has been a wavefront vertex.

Common to all these events is the necessity to compute and store a future arc event whenever two moving intersections become neighbors along an offset circle. The wavefront propagation is active until the highest-weighted input point dominates all other input sites. If multiple sites have the same maximum weight then $\mathcal{Q}$ can only be empty once $\mathcal{W}(S,t)$ contains only one loop of wavefront segments which all lie on offset circles of these sites and if all wavefront vertices move along rays to infinity.

**Theorem 4.1.** Wavefront propagation allows to compute $\mathcal{VD}_w(S)$ in $\mathcal{O}(n^3 \log n)$ time and $\mathcal{O}(n^3)$ space.

*Proof.* Every pair of input sites defines at most constantly many collision and domination events and at most $\mathcal{O}(n^3)$ arc events may take place. Each of these events consumes $\mathcal{O}(\log n)$ time, since every event requires a constant number of lookups, insertions, and/or deletions in a self-balancing binary search tree of size $\mathcal{O}(n)$ or in a priority queue of size $\mathcal{O}(n^3)$. □

We emphasize that the bound $\mathcal{O}(n^3)$ on the number of arc events is a trivial upper bound given by the fact that every triple of offset circles can define only a constant number of arc events. Unfortunately, no sharper bound on the number of arc events is known even just for weighted point sites.

## 5 Variable-Radius Skeleton

Assume that the weighted points and straight-line segments of $S$ form the vertices and edges of a polygon $P$. We call a vertex $v$ *reflex* (*convex*) if the interior angle at $v$ is greater (smaller, resp.) than 180°. (For the sake of descriptional simplicity we assume that no interior angle equals 180°.) Recall that a (positive) weight $w(v)$ is assigned to every vertex $v$. A vertex is *dominant* if its weight is greater than the weight of one of its two neighbors.

Let $A$ denote (the closure of) the area bounded by $P$. We now explain how we obtain a Voronoi-like structure within $A$ that we named variable-radius skeleton (VRS), $\mathcal{S}_w(P)$; see Figure 6. Similar to a Voronoi diagram, it subdivides the interior of $P$ into several *skeletal regions*, with at most one region $\mathcal{SR}_w(s,S)$ per edge $s$ or vertex $s$, for all $s \in S$. However, unlike the GWVD, which may be defined based on an explicit distance function, we rely on a purely wavefront-based definition to specify the VRS such that the following two properties hold:

- For every $s \in S$ its skeletal region $\mathcal{SR}_w(s,S)$ is connected (or empty).
- A point $p$ lies in $\mathcal{SR}_w(s,S)$ if and only if $s \in S$ is the closest site for which there exists a path $\gamma$ within $A$ from $p$ to $s$ such that the weighted distance decreases strictly monotonically as one moves from $p$ to $s$ along $\gamma$.

We will be using a slightly different type of wavefront than in the case of the GWVD. To make this distinction more clear we refer to this new wavefront as the *extinguishing wavefront* $\mathcal{EW}(P,t)$ of $P$ at time $t$. For $t := 0$ we have $\mathcal{EW}(P,t) = P$, i.e., the extinguishing wavefront coincides with $P$. For a sufficiently small time $t_0$, a counter-clockwise traversal of $\mathcal{EW}(P,t_0)$ will match a counter-clockwise traversal of $P$ such that every straight-line segment of $\mathcal{EW}(P,t_0)$ corresponds to exactly one edge of $P$ and every circular arc of $\mathcal{EW}(P,t_0)$ corresponds to exactly one reflex vertex of $P$, and vice versa.

As in the case of a GWVD, the expansion of the wavefront $\mathcal{EW}(P,t)$ is simulated by continuously increasing the time (or weighted distance) $t$. We know that wavefronts are formed by (possibly several) closed curvilinear chains that consist of straight-line segments and circular arcs as wavefront edges. The key difference to the standard wavefront propagation is given by the fact that parts of the wavefront extinguish each other once they collide. In the prairie fire analogy, this means that even a rapidly expanding fire front stops expanding once it reaches an area of $A$ that had already been burnt by another fire. The combinatorial changes that $\mathcal{EW}(P,t)$ can undergo are witnessed by the following three types of events; see Figure 7.

**Definition 5.1** (Split event)**.** A *split event* occurs at time $t$ if an arc of $\mathcal{EW}(P,t)$ collides with another arc or a segment of $\mathcal{EW}(P,t)$, thereby splitting $\mathcal{EW}(P,t')$ into two wavefront components for $t' > t$.

**Definition 5.2** (Edge event)**.** An *edge event* occurs at time $t$ if the two end-points of an edge $e$ of $\mathcal{EW}(P,t)$ coincide in $\mathcal{EW}(P,t)$, thus causing $e$ to disappear and to be absent from $\mathcal{EW}(P,t')$ for $t' > t$.

**Definition 5.3** (Break-through event)**.** A *break-through event* occurs at time $t$ if, for a dominant convex vertex $v_i$ of $P$, the wavefront edges induced by the polygon edges $\overline{v_{i-1}v_i}$ and $\overline{v_i v_{i+1}}$ are collinear at time $t$. In this case $\mathcal{EW}(P,t')$ will contain a circular arc centered at $v_i$ for $t' > t$.

That is, a break-through event witnesses the birth of a new skeletal region and causes a new circular arc to appear on the wavefront. Elementary mathematics shows that a break-through event can occur at most once per convex vertex. An edge event may occur as one offset circle dominates a second one. The locations of edge events and break-through events coincide with nodes of $\mathcal{S}_w(P)$. At an edge event one of the faces induced by $\mathcal{S}_w(P)$ is closed.

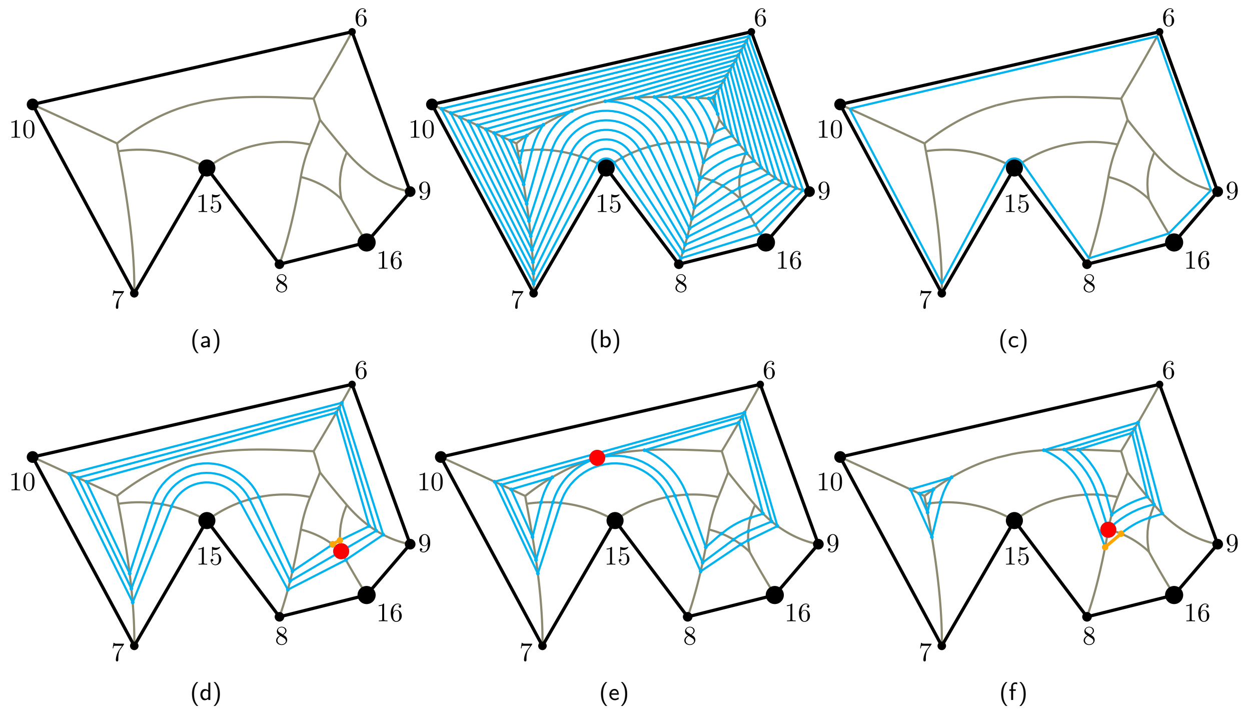


**Figure 7**: (a) Sample variable-radius skeleton (VRS) of a polygon and (b) a family of wavefronts. The numbers next to the vertices of the polygon indicate their weights. (c) Initial wavefront; (d) break-through event; (e) split event; (f) edge event. The location of each event is marked by a red dot, and the wavefront edge that appears or disappears is drawn in orange. The radii of the disks centered at the polygon vertices are directly proportional to the weights that are associated with them. Note that after the split event that is depicted in (e) the wavefront consists of two separate wavefront components.

This event-based description of the VRS already implies a simple construction strategy. All events are stored in a priority queue $\mathcal{Q}$ ordered by the times of their occurrences. Every wavefront component is a curvilinear chain oriented counter-clockwise. It is represented by a doubly-linked list. Every wavefront edge holds a pointer to its predecessor and its successor in the corresponding wavefront component. Additionally, every input site stores the wavefront edges which it is associated with in a self-balancing binary search-tree in counter-clockwise order.

Initially, the times of all potential split events as well as all break-through events are computed and pushed to $\mathcal{Q}$. Furthermore, for every wavefront segment that shrinks to zero length an edge event is inserted into $\mathcal{Q}$.

After the initialization phase is completed the individual events are successively popped from $\mathcal{Q}$. Assume that $\sigma$ is the current event that takes place at time $t_\sigma > 0$.

- If $\sigma$ is a split event then we determine the two wavefront segments $e_1$ and $e_2$ along which the event point $p_\sigma$ is situated and split the corresponding wavefront component accordingly. Thus, a split event involves constantly many pointer manipulations and insertions/deletions in the corresponding self-balancing binary search-trees.
- If $\sigma$ is an edge event then we remove the corresponding wavefront segment from its respective wavefront component. Special precautions have to be taken whenever a wavefront segment disappears since one offset circle $c(s_2, t)$ dominates another offset circle $c(s_1, t)$ and only one of the disappearing moving intersections is currently a wavefront vertex; see Figure 8. In this case, we know that $w(s_1) < w(s_2)$. We boost the weight of $s_1$ to $w(s_2)$ and continue the wavefront propagation. Hence, we will refer to such an edge event as a *boost event*. (This strategy is inherited from a similar situation that can occur for weighted straight skeletons [9].) In order to make sure that we will not miss a split event we have to compute all collisions of the offset circle of $s_1$ with the other wavefront edges of its wavefront component.
- Otherwise, $\sigma$ is a break-through event. We insert a new wavefront edge $e$ at $p_\sigma$ into the respective wavefront component. Again, we check whether $e$ may have future collisions wavefront edges of its wavefront component. In such a case we insert the corresponding split event into $\mathcal{Q}$.

Common to all these events is the need to watch for edge events: Whenever two wavefront vertices become neighbors for the first time we check whether they will coincide at a future point in time. In such a case we insert the corresponding edge event into $\mathcal{Q}$.

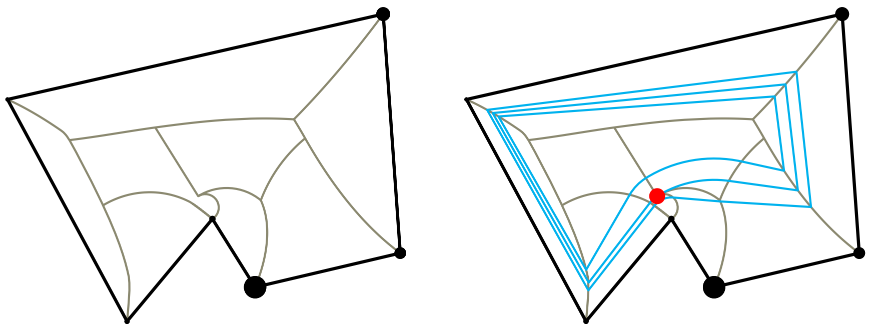


**Figure 8**: A boost event takes place along the wavefront (marked by the red dot).

The wavefront propagation terminates once all wavefront components have shrunk to points and, thus, vanished. Recall that the skeletal region of a site $s$ is swept by portions of the offset circle of $s$. An inductive argument over all event times shows that every skeletal region does indeed have the desired properties. In particular, it is connected. With minor modifications the wavefront propagation used to compute a VRS of weighted points and straight-line segments can be extended to any set of weighted points, weighted straight-line segments and constantly-weighted circular arcs as input sites: We only need to demand that no pair of input sites intersects in a point other than a common end-point.

The attentive reader may wonder why we do not need to deal with break-through and boost events also in the case of GWVDs. After all, the input sites that define a GWVD could also represent the vertices and edges of a polygon. The key difference is given by the fact that a GWVD is computed within the entire plane. This implies that a boost event during the construction of a VRS would manifest itself as a domination event in the case of GWVDs, which causes the wavefront arc whose weight is boosted in the VRS to disappear completely from the wavefront of the GWVD. During a break-through event three moving intersections coincide in a single point. Thus, all break-through events are implicitly dealt with at arc events during the construction of the GWVD.

**Lemma 5.1.** Let $P$ be a polygon with $n$ vertices. Then VRS $\mathcal{S}_w(P)$ has a combinatorial complexity of $\mathcal{O}(n)$ in the worst case.

*Proof.* Recall that we have exactly one skeletal region for each straight-line segment and for each reflex vertex, and at most one region for each convex vertex. Hence, the number of skeletal region is linear in $n$. Furthermore, $\mathcal{S}_w(P)$ is a plane graph where every node that does not coincide with a vertex of $P$ has degree (at least) three. Euler's formula for planar graphs implies that also the number of nodes and edges of $\mathcal{S}_w(P)$ is linear in $n$. $\square$

**Theorem 5.1.** Wavefront propagation allows to compute the variable-radius skeleton of a polygon $P$ with $n$ vertices in worst-case time $\mathcal{O}(nr^2 + nr \log n)$ and $\mathcal{O}(nr)$ space, where $r$ is the total number of reflex and dominant vertices of $P$.

*Proof.* Initially, $\mathcal{O}(nr)$ split events have to be computed. Additionally, $\mathcal{O}(r)$ break-through events may take place in the worst case. Every break-through event takes $\mathcal{O}(n)$ time since we need another round of checks for future split events. During every split and break-through event constantly many wavefront edges are generated that disappear at subsequent edge events. Up to $\mathcal{O}(r^2)$ boost events can occur, with one event consuming $\mathcal{O}(n)$ time due to a search for future split events. All other events consume at most $\mathcal{O}(\log n)$ time, since they require a constant number of lookups, insertions, and/or deletions in a self-balancing binary search tree of size $\mathcal{O}(n)$ or in a priority queue of size $\mathcal{O}(nr)$. $\square$

## 6 Implementation and Discussion

We have been working on a prototype implementation of the wavefront-based construction strategy for variable-radius skeletons in C++. Our implementation operates entirely in two dimensions and is based on conventional IEEE 754 floating-point arithmetic. The skeletons and offsets shown in Figure 11 and the other figures of this publication were generated by means of our implementation. We performed runtime tests on approximately 600 weighted polygons with $n \in \{16, 32, 64, \ldots, 8192\}$ vertices; see Figure 9. (Larger polygons could not be tested due to memory constraints.) All weights were chosen randomly in which we allowed the highest weight to be at most ten-times as big as the lowest one. The polygons were taken from real-world input provided by companies and from the Salzburg Database of polygonal data [8]. In our tests the number $r$ of reflex and dominant vertices averaged about $0.8n$. All tests were carried out on an Intel Xeon E5-2687W v4 processor clocked at 3.0 GHz. These tests support our conjecture that the super-quadratic time bound given in Theorem 5.1 is far too pessimistic for practical applications.

However, the bound $O(r^2)$ on the number of boost events is sharp in the worst case: One can specify a polygon and select appropriate vertex weights such that a cascade of boost events occurs, where the weights of $i-1$ circular arcs of the wavefront get boosted during the $i$-th boost event. Such a setting is highly contrived, though! Our test runs let us conclude that one is more likely to see no boost event at all than to see even just a few such events during one wavefront propagation.

Additionally, we performed several sample test runs in which we bounded the difference between the highest as well as the lowest weighted vertex by a small constant factor, or assigned gradually increasing weights to the vertices we encountered whilst walking along the boundary of the respective input polygon; see Figure 10. Our tests indicate that these special weight-assignment strategies have no significant impact on the overall performance of our implementation (compared to our standard weight-assignment strategy). In particular, the respective runtimes were well within the fluctuation range that we encountered during our standard test runs.

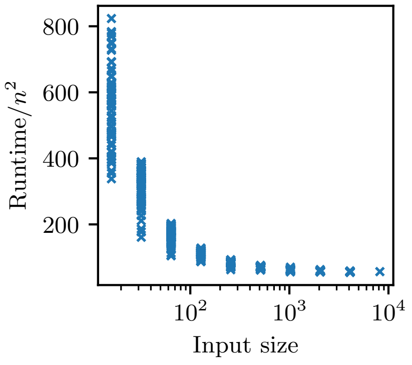


**Figure 9**: The plot shows the overall runtime in microseconds divided by $n^2$. Each marker on the $x$-axis indicates the number $n$ of input vertices for one out of roughly 600 test polygons.

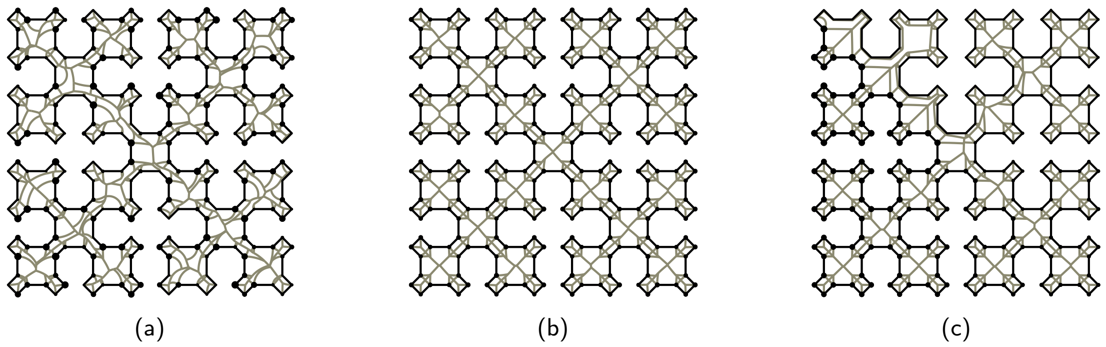


**Figure 10**: Three VRSs inside differently weighted instances of a Sierpinski curve are shown. In (a) and (b) the corresponding vertex weights have been chosen uniformly at random in which the highest and lowest weight are within a factor of 10 (for (a)) and 1.2 (for (b)) of each other. In (c) the vertex weights gradually increase as we walk along the boundary of the input polygon.

The high algebraic degree of the individual bisectors turns the reliable determination of edge events into a delicate problem on conventional floating-point arithmetic. It turned out to be particularly difficult to deal with multiple events that happen at almost the same location. Experience drawn from several industrial-strength implementations of geometric codes carried out within the first author's group during the last 30 years allowed us to mitigate the numerical problems. Still, we have seen our code produce obviously incorrect results due to numerical stability problems. Fortunately, random sampling of a few nodes of a VRS followed by distance computations allows to detect incorrect structures fairly reliably. (And incorrect offsets are also spotted easily by a human by simple visual inspection.) A potential solution to this problem would be to utilize exact arithmetic, e.g., by using the Computational Geometry Algorithms Library (CGAL) [2]. However, exact arithmetic has a hefty computational price tag. In particular, comparisons of event times can no longer be assumed to take (close to) constant time [9]. (Note that we would require the use of constructors and could not resort to a realization based entirely on predicates.)

From an algorithmic point of view a natural avenue for future research is to try to reduce the number of collision/split events computed. Experiments quickly show that many of the quadratically many split events that are computed a-priori are irrelevant. That is, they have no influence on the subsequent wavefront propagation. Thus, it would greatly improve the practical performance of our strategy if we were able to filter out at least some of the unnecessary event computations beforehand. This remark is particularly true if one would resort to exact arithmetic to schedule all events.

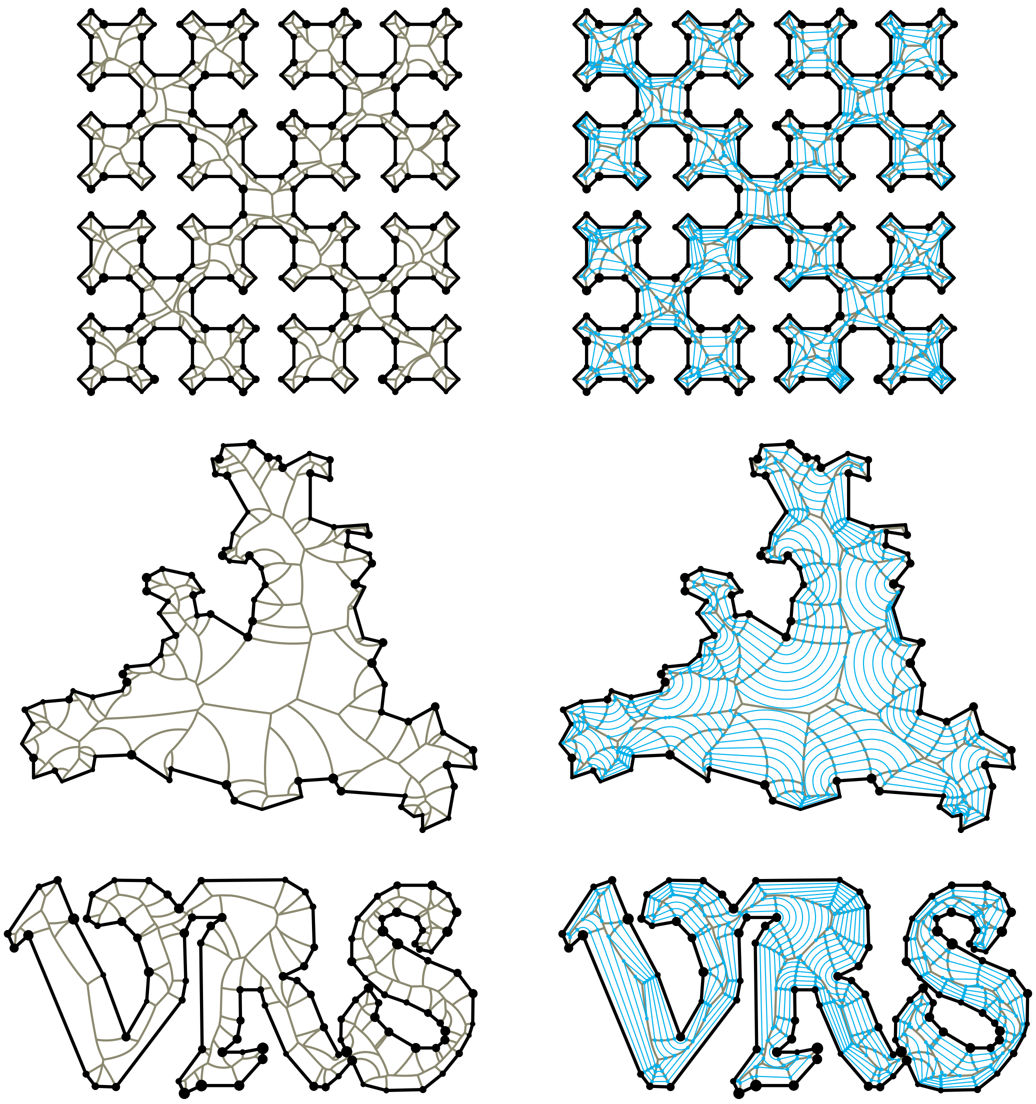


**Figure 11**: Variable-radius skeletons and families of variable-radius offsets.

## 7 Constructing Roofs

Similar to other skeletal structures [13], it is also possible to derive three-dimensional terrains from a VRS that can be used as intricate roofs over polygonal footprints of buildings. For a polygonal footprint $P$, a point $p$ of

$\mathcal{S}_w(P)$, with coordinates $(p_x, p_y)$, is lifted to a point in $\mathbb{R}^3$ with coordinates $(p_x, p_y, d_w(p, s))$. Alternatively, we can lift a wavefront $\mathcal{EW}(P,t)$ for the weighted distance $t$ to $z$-coordinate $t$: We get $\mathcal{EW}(P,t) \times \{t\}$.

We refer to such a roof as a *variable-radius roof* and note that it is guaranteed to drain water. (That is, it does not contain local minima that form sinks in which water would accumulate.) The individual faces of a variable-radius roof are given by parts of conics and by ruled surfaces. Figure 12 illustrates examples of such roofs for some of the polygons and skeletons shown in Figure 11.

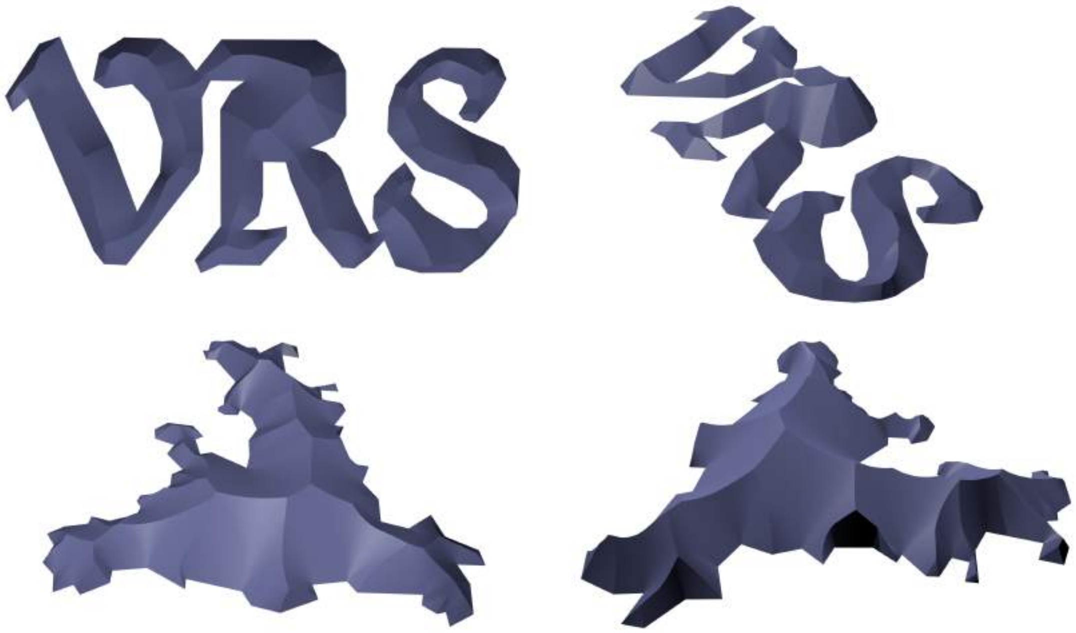


**Figure 12**: Two different views of sample variable-radius roofs. The roof data was generated by our implementation and then rendered by means of Blender [1].

## 8 Conclusion

We extend the work by Held et al. [12] and present a wavefront-based construction strategy for generalized weighted Voronoi diagrams. The obvious practical problem of a GWVD is that a Voronoi region may consist of several connected components even if the edges of $S$ form a polygon $P$. Therefore, the GWVD of a polygon $P$ may have a quadratic combinatorial complexity in the worst case. If such a structure is used to generate ornamental seams then we would get seam curves at locations inside of $P$ where one would not expect to see them. Similarly, the faces of variable-radius roofs would hardly match one's intuition. To remedy this shortcoming, we introduce the variable-radius skeleton inside a polygon $P$. It allows to locally control the spacing of consecutive offset curves by associating multiplicative weights with the vertices of $P$ while still maintaining connected skeletal regions. Our implementation shows the complexity bounds derived for the computation of a variable-radius skeleton tend to be too pessimistic in practical applications. An extension of our strategy to three dimensions would be possible from a purely theoretical point of view. However, an actual implemention of such an extension should be assumed to be difficult in practice.

**Acknowledgements**

Work supported by Austrian Science Fund (FWF): Grant P31013-N31.

## ORCID

*Martin Held,* http://orcid.org/0000-0003-0728-7545
*Stefan de Lorenzo,* http://orcid.org/0000-0003-4981-805X

## REFERENCES

[1] Blender. http://www.blender.org/.

[2] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org/.

[3] Agarwal, P.K.; Schwarzkopf, O.; Sharir, M.: The Overlay of Lower Envelopes and its Applications. Discrete & Comp. Geom., 15(1), 1–13, 1996. http://doi.org/10.1007/BF02716576.

[4] Aurenhammer, F.; Edelsbrunner, H.: An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane. Pattern Recognition, 17(2), 251–257, 1984. http://doi.org/10.1016/0031-3203(84)90064-5.

[5] Biedl, T.; Held, M.; Huber, S.; Kaaser, D.; Palfrader, P.: Weighted Straight Skeletons in the Plane. Comput. Geom. Theory and Appl., 48(2), 120–133, 2015. http://doi.org/10.1016/j.comgeo.2014.08.006.

[6] Dehne, F.; Klein, R.: "The Big Sweep": On the Power of the Wavefront Approach to Voronoi Diagrams. Algorithmica, 17(1), 19–32, 1997. http://doi.org/10.1007/BF02523236.

[7] Edelsbrunner, H.; Seidel, R.: Voronoi Diagrams and Arrangements. Discrete & Comp. Geom., 1(1), 25-44, 1986. http://doi.org/10.1007/BF02187681.

[8] Eder, G.; Held, M.; Jasonarson, S.; Mayer, P.; Palfrader, P.: On Generating Polygons: Introducing the Salzburg Database. In Proc. 36th Europ. Workshop Comput. Geom., 75:1–75:7, 2020.

[9] Eder, G.; Held, M.; Palfrader, P.: On Implementing Straight Skeletons: Challenges and Experiences. In Proc. 36th Int. Sympos. Comput. Geom. (SoCG'20), 38:1–38:16, 2020.

[10] Held, M.: Voronoi Diagrams and Offset Curves of Curvilinear Polygons. Comput. Aided Design, 30(4), 287–300, 1998. http://doi.org/10.1016/S0010-4485(97)00071-7.

[11] Held, M.: VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments. Comput. Geom. Theory and Appl., 18(2), 95–123, 2001. http://doi.org/10.1016/S0925-7721(01)00003-7.

[12] Held, M.; Huber, S.; Palfrader, P.: Generalized Offsetting of Planar Structures using Skeletons. Comput. Aided Design & Appl., 13(5), 712–721, 2016. http://doi.org/10.1080/16864360.2016.1150718.

[13] Held, M.; Palfrader, P.: Skeletal Structures for Modeling Generalized Chamfers and Fillets in the Presence of Complex Miters. Comput. Aided Design & Appl., 16(4), 620–627, 2019. http://doi.org/10.14733/cadaps.2019.620-627.

[14] Klein, R.; Langetepe, E.; Nilforoushan, Z.: Abstract Voronoi Diagrams Revisited. Comput. Geom. Theory and Appl., 42(9), 885–902, 2009. http://doi.org/10.1016/j.comgeo.2009.03.002.

[15] Klein, R.; Mehlhorn, K.; Meiser, S.: Randomized Incremental Construction of Abstract Voronoi Diagrams. Comput. Geom. Theory and Appl., 3(3), 157–184, 1993. http://doi.org/10.1016/0925-7721(93)90033-3.

[16] Palfrader, P.; Held, M.; Huber, S.: On Computing Straight Skeletons by Means of Kinetic Triangulations. In Proc. 20th Annu. Europ. Symp. Algorithms (ESA'12), 766–777, 2012. http://doi.org/10.1007/978-3-642-33090-2_66.