



# A simple algorithm for computing positively weighted straight skeletons of monotone polygons <sup>☆</sup>



Therese Biedl <sup>a,\*</sup>, Martin Held <sup>b</sup>, Stefan Huber <sup>c</sup>, Dominik Kaaser <sup>b</sup>,  
Peter Palfrader <sup>b</sup>

<sup>a</sup> David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 1A2, Canada

<sup>b</sup> Universität Salzburg, FB Computerwissenschaften, 5020 Salzburg, Austria

<sup>c</sup> Institute of Science and Technology Austria, 3400 Klosterneuburg, Austria

## ARTICLE INFO

### Article history:

Received 2 January 2014

Received in revised form 2 September 2014

Accepted 24 September 2014

Available online 30 September 2014

Communicated by R. Uehara

### Keywords:

Computational geometry

Weighted straight skeleton

Monotone polygon

Lower envelope

## ABSTRACT

We study the characteristics of straight skeletons of monotone polygonal chains and use them to devise an algorithm for computing positively weighted straight skeletons of monotone polygons. Our algorithm runs in  $\mathcal{O}(n \log n)$  time and  $\mathcal{O}(n)$  space, where  $n$  denotes the number of vertices of the polygon.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).

## 1. Introduction

The straight skeleton  $\mathcal{S}(\mathcal{P})$  of a simple polygon  $\mathcal{P}$  was introduced by Aichholzer et al. [1] and is defined by considering the propagation of a so-called wavefront. Each edge of  $\mathcal{P}$  emits a wavefront-edge moving at unit speed towards the polygon's interior in a self-parallel manner. During this propagation process, the topology of the wavefront changes due to self-interaction: (i) In an *edge event* an edge of the wavefront shrinks to zero length and thus vanishes. (ii) A *split event* happens when a vertex of the wavefront moves into the interior of a non-incident wavefront edge. (iii) For input that is not in general position

even more complex interactions such as *vertex-events* or *multi-split-events* are possible [2,3]. The straight skeleton is the union of the traces of wavefront vertices over the entire time of the wavefront propagation, see Fig. 1.

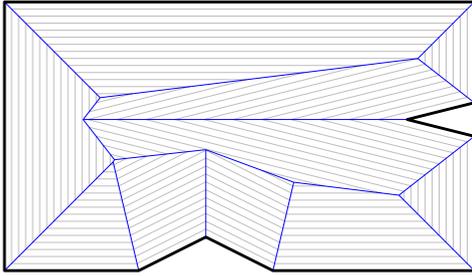
The weighted version of the straight skeleton, where edges no longer move at unit speed, was first mentioned by Eppstein and Erickson [2] and studied in detail by Biedl et al. [4,5]. Several algorithms are known for constructing unweighted straight skeletons, such as those by Aichholzer et al. [1], Eppstein and Erickson [2], Cheng and Vigneron [6], Huber and Held [3], or Vigneron and Yan [7].

Das et al. [8] suggested an algorithm for constructing the (unweighted) straight skeleton of monotone polygons, which they claim runs in  $\mathcal{O}(n \log n)$  time, where  $n$  denotes the number of vertices of the polygon. However, we have simple examples that show that their Lemmas 5, 6, and 7 do not hold for all valid inputs. In particular, their approach hinges upon the assumption that no event introduces a new reflex vertex during the wavefront propagation process, which is clearly incorrect for general input. (See the node marked in green in Fig. 2, on the right-hand

<sup>☆</sup> Research of Therese Biedl was supported by NSERC and undertaken while visiting the University of Salzburg. Martin Held and Peter Palfrader were supported by Austrian Science Fund (FWF): P25816-N15.

\* Corresponding author.

E-mail addresses: [biedl@uwaterloo.ca](mailto:biedl@uwaterloo.ca) (T. Biedl), [held@cosy.sbg.ac.at](mailto:held@cosy.sbg.ac.at) (M. Held), [stefan.huber@ist.ac.at](mailto:stefan.huber@ist.ac.at) (S. Huber), [dominik@cosy.sbg.ac.at](mailto:dominik@cosy.sbg.ac.at) (D. Kaaser), [palfrader@cosy.sbg.ac.at](mailto:palfrader@cosy.sbg.ac.at) (P. Palfrader).



**Fig. 1.** The straight skeleton  $S(\mathcal{P})$  of an input polygon  $\mathcal{P}$  (bold) is the union of the traces of wavefront vertices. For different points in time the wavefronts are shown in gray.

side of the lower chain.) Note that a perturbation of the input in order to avoid such a vertex event, as suggested by Das et al. [8], cannot be applied as the straight skeleton changes discontinuously [2].

Our algorithm can compute the positively weighted straight skeleton of a monotone polygon in  $\mathcal{O}(n \log n)$  time and  $\mathcal{O}(n)$  space, which constitutes a significant improvement over the  $\mathcal{O}(n^{17/11+\epsilon})$  worst-case time and space complexity of the currently best algorithm for arbitrary simple polygons by Eppstein and Erickson [2]. (The algorithm by Vigneron and Yan [7] achieves an expected  $\mathcal{O}(n^{4/3} \log n)$  time complexity but is only applicable if no multi-split events occur.) Furthermore, our algorithm does not require complex data structures and is easy to implement.

## 2. Strictly monotone polygonal chains

Let  $\mathcal{C}$  be a polygonal chain strictly monotone with respect to the  $x$ -axis. We construct  $\mathcal{C}'$  as a chain consisting of the same line segments as  $\mathcal{C}$  except that we extend the first and the last segment to rays to the west ( $x \rightarrow -\infty$ ) and the east ( $x \rightarrow +\infty$ ), respectively. Thus,  $\mathcal{C}'$  partitions the plane into an upper and lower portion. In the following we continue to call such an unbounded chain a polygonal chain.

We start by considering the unweighted wavefront propagation of  $\mathcal{C}'$  where all edges of  $\mathcal{C}'$  emanate a self-parallel wavefront edge towards  $y \rightarrow -\infty$ , which we call the south. (We discuss extensions to positive weights and non-strict monotonicity later on.) We denote the wavefront at time  $t$  by  $\mathcal{W}_{\mathcal{C}}(t)$ . For the initial wavefront, at  $t = 0$ , the chains  $\mathcal{W}_{\mathcal{C}}(t)$  and  $\mathcal{C}'$  are geometrically identical. Again, the first and last segments are rays to infinity. For small values of  $t$  and prior to the first event, the edges of  $\mathcal{W}_{\mathcal{C}}(t)$  are in one-to-one correspondence to the edges of  $\mathcal{C}'$  and occur in the same order.

We denote by  $\mathcal{A}_{\mathcal{C}}(t) = \bigcup_{0 \leq t' \leq t} \mathcal{W}_{\mathcal{C}}(t')$  the area swept by the wavefront until time  $t$ . The *roof model*  $\mathcal{T}(\mathcal{C})$  is a three-dimensional structure: We assign to every point of  $\mathcal{A}_{\mathcal{C}}(\infty)$  the time when it was swept by the wavefront. The resulting structure  $\mathcal{T}(\mathcal{C}) = \bigcup_{0 \leq t} (\mathcal{W}_{\mathcal{C}}(t) \times \{t\})$  is a terrain; it helps us in studying the wavefront over its entire propagation period.

**Lemma 1.** *The wavefront  $\mathcal{W}_{\mathcal{C}}(t)$  of a strictly  $x$ -monotone chain  $\mathcal{C}$  is a strictly  $x$ -monotone chain for all times  $t$ . Furthermore,*

*every change in the topology of the wavefront is witnessed by an edge collapsing to zero length.*

**Proof.** We will show this claim by induction: The monotonicity of  $\mathcal{W}_{\mathcal{C}}(t)$  is unaffected by the southwards propagation at least as long as its topology does not change. Since the initial wavefront is identical to  $\mathcal{C}'$ , which is strictly  $x$ -monotone, it follows that  $\mathcal{W}_{\mathcal{C}}(t)$  is also strictly  $x$ -monotone for sufficiently small  $t$  such that no event has happened yet.

The topology of  $\mathcal{W}_{\mathcal{C}}(t)$  can only change as a result of the wavefront interacting with itself, when vertices of  $\mathcal{W}_{\mathcal{C}}(t)$  become incident to other elements of  $\mathcal{W}_{\mathcal{C}}(t)$  which they were previously not incident to. Assume that at a certain point in the propagation process a vertex  $v$  becomes incident to either a different vertex  $v'$  or to an edge  $e$  that it was not previously incident to. If prior to the change the wavefront was a strictly monotone polygonal chain, then it follows that the only way that  $v$  could have become incident to  $v'$  or  $e$  is for all segments between  $v$  and  $v'$  or  $e$  to have shrunk to zero length.

Segments that shrink to zero length in an event are removed from the wavefront. Removing one or more such collapsed segments of a strictly  $x$ -monotone chain and then connecting the remaining pieces together yields a strictly  $x$ -monotone chain again.

Therefore, each event is witnessed by an edge collapse and it transforms one strictly  $x$ -monotone wavefront into another strictly  $x$ -monotone wavefront, resulting in  $\mathcal{W}_{\mathcal{C}}(t)$  being strictly  $x$ -monotone for all times  $t$ .  $\square$

Note that we say that each change is witnessed by an *edge collapse* instead of an *edge event*. We do this in order to also include non-elementary changes of the wavefront, such as vertex events.

**Theorem 2.** *The straight skeleton  $S(\mathcal{C})$  of a strictly monotone polygonal chain  $\mathcal{C}$  can be computed in time  $\mathcal{O}(n \log n)$ , where  $n$  is the number of edges of  $\mathcal{C}$ .*

**Proof.** We maintain a priority queue to keep track of the edge collapses, which witness all topological changes. The total number of propagating wavefront edges initially is  $n$ . We can compute the initial collapse times of all these edges in  $\mathcal{O}(n)$  time and fill a priority queue in total  $\mathcal{O}(n \log n)$  time.

We keep fetching the next edge collapse from the priority queue. We have to re-compute the collapse times of the two incident edges and adapt their entries in the priority queue in  $\mathcal{O}(\log n)$  time. Collapse times of other edges are not affected by such a change. Note that the number of wavefront edges decreases with each event because no split events occur.

Since all topological changes of the wavefront are witnessed by such edge collapses (Lemma 1), no event is missed. Therefore, we can construct the straight skeleton  $S(\mathcal{C})$  in total  $\mathcal{O}(n \log n)$  time.  $\square$

**Lemma 3.** *Each wavefront edge of  $\mathcal{W}_{\mathcal{C}}(t)$  has area already swept by the wavefront to its immediate north and unswept area to its south.*

**Proof.** The part of the plane that has already been swept,  $\mathcal{A}_C(t)$ , lies between  $C'$  and  $\mathcal{W}_C(t)$ . Since  $\mathcal{W}_C(t)$  is strictly  $x$ -monotone, each element of  $\mathcal{W}_C(t)$  has swept area to its north and unswept area to its south.  $\square$

**Lemma 4.** Let  $\mathcal{T}(C)$  be the roof of a strictly monotone chain  $C$  that emanated a wavefront southwards. Then, on any point on the surface of  $\mathcal{T}(C)$ , the elevation of the roof increases when moving due south.

**Proof.** This holds for each individual face of  $\mathcal{T}(C)$  because of Lemma 3. Since the roof is continuous, it also holds when moving between faces.  $\square$

### 3. Strictly monotone polygons

To compute the unweighted straight skeleton of a strictly monotone polygon  $\mathcal{P}$ , which we assume to be monotone with respect to the  $x$ -axis, we split  $\mathcal{P}$  into two chains, the northern or top (upper) chain  $C_u$  and the southern or bottom chain  $C_b$ . The common western and eastern vertices are denoted by  $v_w$  and  $v_e$ , respectively. Both chains of  $\mathcal{P}$  emanate their wavefronts inwards, that is,  $C_u$  emanates its wavefront southwards,  $C_b$  northwards. We compute the straight skeleton and roof for both chains independently and denote these by  $\mathcal{S}(C_u)$ ,  $\mathcal{S}(C_b)$ ,  $\mathcal{T}(C_u)$ ,  $\mathcal{T}(C_b)$ .

**Definition 1.** Let  $\mathcal{M}$  be a polygonal chain in  $\mathbb{R}^3$ . We say  $\mathcal{M}$  is strictly 3D-monotone with respect to the  $x$ -axis if every plane parallel to the  $yz$ -plane intersects  $\mathcal{M}$  in at most one point.

**Lemma 5.** Let  $\mathcal{M}$  be a polygonal chain in  $\mathbb{R}^3$  that is strictly 3D-monotone with respect to the  $x$ -axis. Then projecting  $\mathcal{M}$  onto the  $xy$ -plane yields a polygonal chain  $\mathcal{M}'$  that is strictly  $x$ -monotone.

**Proof.** Let  $h$  be an arbitrary line in the  $xy$ -plane orthogonal to the  $x$ -axis (i.e., parallel to the  $y$ -axis). To establish that  $\mathcal{M}'$  is  $x$ -monotone, we need to show that  $h$  intersects  $\mathcal{M}'$  in at most one point.

Let  $H$  be a plane parallel to the  $yz$ -plane such that  $h$  lies within  $H$ . By assumption,  $H$  intersects  $\mathcal{M}$  in at most one point. Thus, the projection of  $\mathcal{M}$  onto the  $xy$ -plane intersects  $h$  in at most one point.  $\square$

**Lemma 6.** The two roofs  $\mathcal{T}(C_u)$  and  $\mathcal{T}(C_b)$  intersect in a polygonal chain  $\mathcal{M}$  in  $\mathbb{R}^3$  that is strictly 3D-monotone with respect to the  $x$ -axis. This merge chain starts in  $v_w$  and ends in  $v_e$ .

**Proof.** Monotonicity of the merge chain  $\mathcal{M}$  follows from Lemma 4: Let  $\Pi$  be an arbitrary plane parallel to the  $yz$ -plane that intersects  $\mathcal{W}_{C_b}(0)$  in  $p_b$  and  $\mathcal{W}_{C_u}(0)$  in  $p_u$ . If  $\Pi$  intersects the polygon  $\mathcal{P}$  then  $p_b$  is the southern intersection and  $p_u$  the northern intersection of  $\Pi$  and  $\mathcal{P}$ .

The intersection of  $\Pi$  with  $\mathcal{T}(C_b)$  is a piecewise linear terrain function in  $\Pi$  that starts at point  $p_b$ , where  $z = 0$ , and strictly monotonically increases its  $z$ -coordinate as the  $y$ -coordinate increases towards  $y \rightarrow +\infty$ . Similarly,

the intersection of  $\Pi$  with  $\mathcal{T}(C_u)$  is a strictly monotonically decreasing function starting at  $y \rightarrow -\infty$  and positively infinite  $z$ -coordinate and ending at point  $p_u$  where  $z = 0$ .

Two such functions coincide in exactly one point if  $p_u$  is north of  $p_b$ , as is the case if  $\Pi$  intersects the polygon. If the intersection of  $\Pi$  and  $\mathcal{P}$  is empty, then  $p_b$  is north of  $p_u$  and the two terrain functions do not intersect at all. Therefore,  $\mathcal{M}$  is strictly 3D-monotone with respect to the  $x$ -axis.

Vertices  $v_w$  and  $v_e$  are the start and end points of the chain because they are the common vertices of both chains. Since both roofs consist of planar faces, the intersection consists of line segments.  $\square$

In the merge step, we construct a new polyhedron  $\mathcal{T}$  by stitching together the faces of  $\mathcal{T}(C_u)$  between  $C_u$  and  $\mathcal{M}$  and the faces of  $\mathcal{T}(C_b)$  between  $C_b$  and  $\mathcal{M}$ . Note that this polyhedron is a terrain above the interior of  $\mathcal{P}$  and its intersection with the  $xy$ -plane is equal to  $\mathcal{P}$ . Furthermore,  $\mathcal{T}$  is piecewise-linear and continuous and each face is incident to one edge of  $\mathcal{P}$ . It remains to show that this roof  $\mathcal{T}$  is equivalent to the straight-skeleton induced roof  $\mathcal{T}(\mathcal{P})$  of  $\mathcal{P}$ .

**Lemma 7.** All edges introduced by our merge step, i.e., the edges of  $\mathcal{M}$ , are ridges in  $\mathcal{T}$ , not valleys.

**Proof.** Each edge  $e$  of  $\mathcal{M}$  is incident to one face  $f_u$  of the northern roof on its north side, and incident to one face  $f_b$  of the southern roof to its south. By Lemma 4,  $f_u$  is sloping downwards towards north, and  $f_b$  is sloping downwards towards south. Thus,  $e$  is a ridge.  $\square$

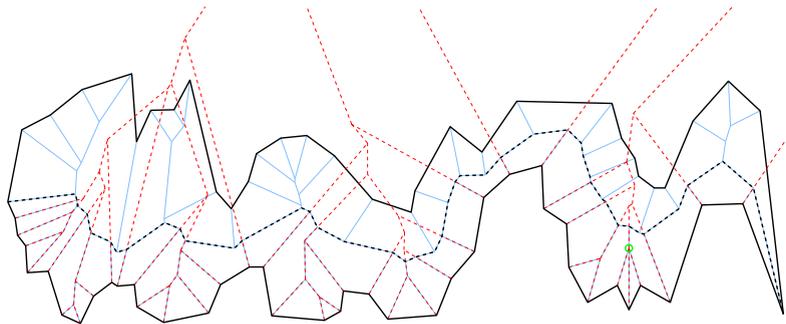
**Theorem 8.** Let  $\mathcal{T}$  be the roof constructed by merging  $\mathcal{T}(C_u)$  and  $\mathcal{T}(C_b)$  as described. Then  $\mathcal{T}$  is the roof induced by  $\mathcal{S}(\mathcal{P})$ .

**Proof.** In the following we will only consider the roof above  $\mathcal{P}$ . Note that  $\mathcal{T}$  is the unique lower envelope of  $\mathcal{T}(C_u)$  and  $\mathcal{T}(C_b)$ . Let  $\mathcal{T}(\mathcal{P})$  denote the roof induced by  $\mathcal{S}(\mathcal{P})$ . We need to show that  $\mathcal{T}(\mathcal{P})$  also is the lower envelope of  $\mathcal{T}(C_u)$  and  $\mathcal{T}(C_b)$ .

As  $\mathcal{S}(\mathcal{P})$  is a tree, there is a unique path  $\mathcal{M}'_{\mathcal{S}}$  between  $v_w$  and  $v_e$ . This is exactly the path that separates the union of faces incident to  $C_u$  and the union of faces incident to  $C_b$ . Hence,  $\mathcal{M}'_{\mathcal{S}}$  comprises exactly the straight-skeleton arcs that have faces of different chains on either side. From that it follows that  $\mathcal{M}'_{\mathcal{S}}$  is  $x$ -monotone.

We build vertical slabs above the edges of  $\mathcal{M}'_{\mathcal{S}}$ , resulting in the intersection  $\mathcal{M}_{\mathcal{S}}$  with  $\mathcal{T}(C_b)$ . All straight-skeleton nodes of  $\mathcal{S}(\mathcal{P})$  south of  $\mathcal{M}'_{\mathcal{S}}$  originate from topological changes within  $C_b$ . Hence, their lifted counterparts in  $\mathcal{T}(\mathcal{P})$  coincide with vertices of  $\mathcal{T}(C_b)$ . Likewise, every vertex in  $\mathcal{T}(C_b)$  south of  $\mathcal{M}_{\mathcal{S}}$  has its counterpart as a node of  $\mathcal{S}(\mathcal{P})$  south of  $\mathcal{M}'_{\mathcal{S}}$ . In other words,  $\mathcal{T}(C_b)$  and  $\mathcal{T}(\mathcal{P})$  coincide south of  $\mathcal{M}'_{\mathcal{S}}$ . With the same argument,  $\mathcal{T}(C_u)$  and  $\mathcal{T}(\mathcal{P})$  coincide north of  $\mathcal{M}'_{\mathcal{S}}$ . In particular,  $\mathcal{M}_{\mathcal{S}} = \mathcal{T}(C_b) \cap \mathcal{T}(C_u)$ .

Finally, we observe that every path on  $\mathcal{T}(\mathcal{P})$  from  $\mathcal{M}_{\mathcal{S}}$  strictly to the north or the south is descending. On the



**Fig. 2.** The straight skeleton  $\mathcal{S}(\mathcal{P})$  of a strictly monotone input polygon  $\mathcal{P}$  results from the merge of the straight skeletons  $\mathcal{S}(\mathcal{C}_u)$  and  $\mathcal{S}(\mathcal{C}_b)$  of the top and bottom chain –  $\mathcal{S}(\mathcal{C}_b)$  is shown dotted. The merge chain is shown in bold and dotted.

other hand, a path on  $\mathcal{T}(\mathcal{C}_u)$  to the south is ascending. The same is true for paths on  $\mathcal{T}(\mathcal{C}_b)$  to the north. Hence,  $\mathcal{T}(\mathcal{P})$  is indeed the lower envelope of  $\mathcal{T}(\mathcal{C}_u)$  and  $\mathcal{T}(\mathcal{C}_b)$ .  $\square$

**Fig. 2** illustrates the merge by showing a polygon  $\mathcal{P}$  and its straight skeleton  $\mathcal{S}(\mathcal{P})$  as well as the straight skeleton of the bottom chain  $\mathcal{S}(\mathcal{C}_b)$ .

**Corollary 9.** Projecting  $\mathcal{T}$  onto the  $xy$ -plane yields the straight skeleton of the input polygon  $\mathcal{P}$ .

#### 4. Computing the straight skeleton

*Computing the merge chain* We construct the intersection  $\mathcal{M}$  of the northern and southern roofs. The roofs and intersections that appear in this problem have special properties, and we can, thus, find  $\mathcal{M}$  in  $\mathcal{O}(n \log n)$  time.

Since in the end we are only interested in the projection of  $\mathcal{M}$  onto the  $xy$ -plane, we only construct this projection  $\mathcal{M}'$  in 2D. The three-dimensional chain  $\mathcal{M}$  can be extracted from  $\mathcal{M}'$  by raising each vertex by its orthogonal distance to the edges defining its incident faces.

Let  $\mathcal{C}_u$  consist of the edges  $e_{u,1}, e_{u,2}, e_{u,3}, \dots, e_{u,N_u}$ , in order from west to east, and  $\mathcal{C}_b$  of the edges  $e_{b,1}, e_{b,2}, e_{b,3}, \dots, e_{b,N_b}$ . Furthermore, let  $\mathcal{M}'$  consist of vertices and edges  $v_0 = v_w, m_1, v_1, m_2, v_2, m_3, v_3, \dots, v_{N_m-1}, m_{N_m}, v_{N_m} = v_e$ .

We construct  $\mathcal{M}'$  incrementally, starting at  $v_w$  and adding line segments  $m_i$  until we reach  $v_e$ . At each step in the process we keep track of the faces of  $\mathcal{S}(\mathcal{C}_u)$  and  $\mathcal{S}(\mathcal{C}_b)$  that  $m_i$  lies in. We denote these faces by  $f_{u,i}$  and  $f_{b,i}$ . Note that  $m_i$  lies on the bisector of the input edges defining  $f_{u,i}$  and  $f_{b,i}$  since  $\mathcal{M}'$  traces the intersection of the northern and southern roofs.

The initial merge segment  $m_1$  starts at  $v_0 = v_w$ . The northern and southern faces are  $f(e_{u,1})$  and  $f(e_{b,1})$ , where  $f(e)$  denotes the face incident to edge  $e$  in the corresponding straight skeleton. The supporting line of  $m_1$  will intersect arcs of  $\mathcal{S}(\mathcal{C}_u)$  and  $\mathcal{S}(\mathcal{C}_b)$  in loci other than  $v_0$ . Let  $v_1$  be the intersection east of  $v_0$  that minimizes the length of the edge  $m_1 = \overline{v_0 v_1}$ .

Assume  $v_1$  was the intersection of the supporting line of  $m_1$  with an arc  $a$  of the northern straight skeleton  $\mathcal{S}(\mathcal{C}_u)$ . We set the northern face  $f_{u,2}$  to be the other face incident to  $a$ . The southern face does not change and thus

$f_{b,2} = f_{b,1}$ . Should the intersection have occurred for an arc of the southern face, we set  $f_{u,2}$  and  $f_{b,2}$  accordingly. The next segment,  $m_2$ , then lies on the bisector of the input edges defining  $f_{u,2}$  and  $f_{b,2}$ . This segment starts in  $v_1$  and we determine the next vertex  $v_3$  in turn.

Since the segments of  $m_i$  always lie between the northern and southern chain, and since  $\mathcal{M}'$  is monotone, this process will, eventually, end up in  $v_e$  and we will have completed the merge chain.

*Complexity considerations* In order to quickly find the next intersection of  $m_i$  with an arc of either  $\mathcal{S}(\mathcal{C}_u)$  and  $\mathcal{S}(\mathcal{C}_b)$  we compute trapezoidations of  $\mathcal{S}(\mathcal{C}_u)$  and  $\mathcal{S}(\mathcal{C}_b)$  by adding vertical line segments as required.

In the incremental step, when we are looking for  $v_i$ , i.e., the eastern vertex of  $m_i$ , we can use the trapezoidations of both  $f_{u,i}$  and  $f_{b,i}$  to find  $v_i$ : We sweep through the cells from west to east until we find one where the supporting line of  $m_i$  intersects an arc on a cell's boundary, thus finding  $v_i$ . During that switch we enter a new face in either the northern or the southern straight skeleton and we start anew. When searching for  $v_{i+1}$  in this next step, we no longer need to consider the cells that we already visited.

At every stage we intersect  $m_i$  with the boundary of the currently active northern and southern cells to either find  $v_i$  or to move forward in one of the trapezoidations. We can charge each such set of intersection tests to the one cell we move out of. The number of total trapezoidation cells is linear in the input size, and, therefore, given a trapezoidation, the complete merge step can be achieved in time linear in the input size.

A plane sweep allows to compute the trapezoidation of a planar straight-line graph such as  $\mathcal{S}(\mathcal{C}_u)$  in  $\mathcal{O}(n \log n)$  time, where  $n$  is the size of the input graph. Since the size of the straight skeleton is linear in its input, the total cost of creating both trapezoidations therefore also is in  $\mathcal{O}(n \log n)$ .

By **Theorem 2**, the northern and southern straight skeletons can be found in  $\mathcal{O}(n \log n)$  time, and they can also be merged in  $\mathcal{O}(n \log n)$  time. Therefore, the unweighted straight skeleton of a strictly monotone polygon can be computed in  $\mathcal{O}(n \log n)$  time. Obviously, all data structures require space linear in the input size.

## 5. Positively weighted straight skeleton

We now consider a wavefront propagation where not all edges move at the same speed. Recently Biedl et al. [4] showed that many of the seemingly obvious properties of straight skeletons no longer hold when weights are not unit weights. Therefore, diligent consideration is required when extending existing proofs to weighted straight skeletons.

*Monotone chains* None of the proofs of any statement leading up to and including Theorem 2 used the fact that wavefront edges move at unit speed or any properties which depend on unit speeds.

Thus, Theorem 2 still holds and the positively weighted straight skeleton and roof of a polygonal chain can be computed in  $\mathcal{O}(n \log n)$  time.

Since we consider only positive weights, the wavefront of a strictly monotone chain still propagates southwards everywhere. Thus, Lemma 3 and Lemma 4 still hold, and the straight-skeleton induced roof is still a terrain.

*Monotone polygons* For strictly monotone polygons the positively weighted straight skeleton can also be constructed in the way described in Sections 3 and 4. Since Lemma 4 still holds, the intersection of the northern and southern roofs again produces a 3D- $x$ -monotone polygonal chain that consists of ridges only. Therefore, Lemmas 6 and 7 also hold for positively weighted input. Biedl et al. [4] showed that for simple polygons and positive weights the straight skeleton is still a tree. Thus, Theorem 8 still applies and therefore also Corollary 9. Hence, our algorithm is also correct for positively weighted straight skeletons.

## 6. Discussion

We presented an algorithm that computes the positively weighted straight skeleton of a strictly monotone polygon in  $\mathcal{O}(n \log n)$  time and linear space. The restriction to strict monotonicity makes several proofs easier and more readable but can be waived. The obvious problem when dealing with a polygon which, after suitable rotation, is monotone but not strictly monotone relative to the  $x$ -axis is that moving due northwards or southwards in  $\mathcal{T}(\mathcal{C}_u)$  and  $\mathcal{T}(\mathcal{C}_b)$  does no longer guarantee that the elevation increases or decreases, respectively. Furthermore, the merge chain  $\mathcal{M}'_{\mathcal{S}}$  need no longer be strictly monotonous.

We refrain from adding clumsy new proofs to extend our lemmas to this special case. Rather, we content ourselves with noting that suitable subdivisions of the top and bottom chains (in order to isolate edges parallel to the  $y$ -axis within individual chains) followed by suitable (small) rotations allow to obtain the properties claimed by our lemmas locally for the individual pieces of the chains, their roofs, and the final merge chain  $\mathcal{M}'_{\mathcal{S}}$ . Thus, our algorithm can be extended to polygons that are monotone but not strictly monotone, without sacrificing its performance.

An extension of our approach to negative weights seems much more demanding since several important properties are lost. For instance, the roof of a monotone polygonal chain need not be a terrain once negative weights are allowed. For our approach to still work we would need to establish that the northern and southern roofs nevertheless still only intersect in a single, unique merge chain. Note that this merge chain need no longer be monotone, might include vertices at infinity, and its projection onto the  $xy$ -plane might self-intersect. The straight skeleton of a monotone polygon might contain cycles, and even for a convex polygon it may be self-intersecting [4].

## References

- [1] O. Aichholzer, D. Alberts, F. Aurenhammer, B. Gärtner, Straight skeletons of simple polygons, in: Proc. 4th Internat. Symp. of LIESMARS, 1995, pp. 114–124.
- [2] D. Eppstein, J. Erickson, Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions, *Discrete Comput. Geom.* 22 (4) (1999) 569–592.
- [3] S. Huber, M. Held, A fast straight-skeleton algorithm based on motorcycle graphs, *Int. J. Comput. Geom. Appl.* 22 (5) (2012) 471–498, <http://dx.doi.org/10.1142/S0218195912500124>.
- [4] T. Biedl, M. Held, S. Huber, D. Kaaser, P. Palfrader, Weighted straight skeletons in the plane, *Comput. Geom.* 48 (2) (2015) 120–133, <http://dx.doi.org/10.1016/j.comgeo.2014.08.006>.
- [5] T. Biedl, S. Huber, P. Palfrader, Planar matchings for weighted straight skeletons, in: Proc. 25th Annu. Internat. Sympos. Algorithms & Comput., Jeonju, Korea, 2014.
- [6] S.-W. Cheng, A. Vigneron, Motorcycle graphs and straight skeletons, *Algorithmica* 47 (2007) 159–182, <http://dx.doi.org/10.1007/s00453-006-1229-7>.
- [7] A. Vigneron, L. Yan, A faster algorithm for computing motorcycle graphs, in: Proc. 29th Annu. ACM Sympos. Comput. Geom., 2013, pp. 17–26.
- [8] G. Das, A. Mukhopadhyay, S. Nandy, S. Patil, S. Rao, Computing the straight skeleton of a monotone polygon in  $\mathcal{O}(n \log n)$  time, in: Proc. 22nd Canad. Conf. Comput. Geom., 2010, pp. 207–210.