Computing Positively Weighted Straight Skeletons of Simple Polygons Using an Induced Line Arrangement

Günther Eder*[1] and Martin Held†[1]

[1]Universität Salzburg, FB Computerwissenschaften, 5020 Salzburg, Austria

Abstract

We extend the work by Huber and Held (IJCGA 2012) on straight-skeleton computation based on motorcycle graphs to positively weighted skeletons. Resorting to a line arrangement induced by the r reflex vertices of a simple n-vertex polygon P allows to compute the weighted straight skeleton of P in $\mathcal{O}(n^2 + r^3/k + nr \log n)$ time and $\mathcal{O}(n + kr)$ space, for an arbitrary positive integer k with $1 \leq k \leq r$.

Introduction

We consider a simple planar polygon P (without holes) with n vertices and assume that strictly positive weights for the edges of P are given as part of the input. As usual, we call a vertex v reflex if the interior angle at v is greater than π, and convex otherwise. Suppose that r out of the n vertices of P are reflex. We embed P into the xy-plane of $\mathbb{R}^3$.

Wavefront propagation is a well-known strategy for computing (weighted) straight skeletons [2, 3, 6]. The moving wavefront is defined over P and regarded as a function $\mathcal{W}_P(t)$ of time t. At the start time $t = 0$ the wavefront $\mathcal{W}_P(0)$ equals P. As time progresses the propagation process simulates a shrinking of the wavefront. Therefore, 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 out arcs which form the straight skeleton $\mathcal{S}(P)$. To maintain the weak planarity[1] of the wavefront two events have to be handled: edge event and split event. An edge event occurs when a wavefront edge shrinks to zero length. A split event occurs when a reflex wavefront vertex crashes into the interior of an opposing wavefront edge. These events produce the (interior) nodes of the straight skeleton such that at least three arcs meet in a common node. (In addition there are n nodes that correspond to the vertices of P.) If multiple split events occur at the same point and time, i.e., if multiple reflex wavefront vertices coincide, then we call such an event a multi-split event.[2] Multiple edge events on the same point result in a vanishing wavefront component or multiple vanishing edges.

In the weighted scenario every edge requires an additional parameter as the wavefront edges move with speeds induced by edge weights of P rather than with unit speed. We denote by $\mathcal{S}(P, \sigma)$ the weighted straight skeleton of P and by $\mathcal{W}_P(t, \sigma)$ the respective moving wavefront of P, where σ is the edge function that assigns a weight $\sigma(e) > 0$ to every edge e of P. The offset supporting line of the edge e at time t is given by $\overline{e(t)} := \ell(e) + n_e \cdot \sigma(e) \cdot t$, where $\ell(e)$ is the supporting line of e and n_e the inward unit normal vector of e. We let $e(t)$ denote all straight-line edges of $\overline{e(t)}$ that are part of $\mathcal{W}_P(t, \sigma)$ at time t. A wavefront vertex $v(t)$ is defined by the intersection $\overline{e_i(t)} \cap \overline{e_j(t)}$ of two offset supporting lines if an edge of both $e_i(t)$ and $e_j(t)$ is incident at $v(t)$. A reflex (convex) wavefront vertex traces out a *reflex* (*convex*, resp.) *arc* of $\mathcal{S}(P, \sigma)$.

Biedl et al. [3] show that many properties of unweighted straight skeletons are preserved for positively weighted straight skeletons of simple polygons. In particular $\mathcal{S}(P, \sigma)$ is connected, is a tree, has no crossings, and consists of $n + v - 1$ arcs, where v denotes the number of straight skeleton nodes. Furthermore, $\mathrm{cl}(\mathcal{W}_P(t + \varepsilon, \sigma)) \subsetneq \mathrm{cl}(\mathcal{W}_P(t, \sigma))$ for any $\varepsilon > 0$, where cl() denotes the closure of the area bounded by the polygon(s).[3] A property that does not transfer is the monotonicity of a face traced out by $e(t)$ of $\mathcal{W}_P(t, \sigma)$. However, it was shown by Biedl et al. [3] that such a face always forms a simple polygon.

A motorcycle graph, introduced by Eppstein and Erickson [5], is a simulation of r motorcycles $(m_1, \ldots, m_r)$ that have given starting points and velocity vectors in $\mathbb{R}^2$. All motorcycles start at the same time, drive along straight lines at constant speed, and leave traces behind. Whenever one motorcycle

*Email: geder@cosy.sbg.ac.at

†Email: held@cosy.sbg.ac.at

Work supported by Austrian Science Fund (FWF) Grant P25816-N15.

[1]A polygon is weakly planar (or weakly simple) if it is the boundary of a region that is topologically equivalent to a disk; (portions of) edges may overlap and vertices may coincide.

[2]Multi-split events are also known as vertex events [5].

[3]A roof $\mathcal{R}(P)$ can be constructed over P by assigning a time depending z-component to the propagating wavefront, and this roof remains a z-monotone terrain even in the weighted case [3].

crashes into such a trace it stops. The traces remain and form the line segments or rays of a graph: The motorcycle graph $\mathcal{M}(m_1, \dots, m_r)$ is defined as the arrangement of all traces after infinite time.

1 Related Work and Our Contribution

The straight skeleton was introduced two decades ago by Aichholzer et al. [2]. The algorithm with the best worst-case complexity is by Eppstein and Erickson. Their algorithm computes the weighted straight skeleton of a simple polygon (with holes) in $\mathcal{O}(n^{17/11+\varepsilon})$ time and space [5]. Their approach seems challenging to implement, though. More recent results with lower time/space-complexity are known [4, 8]. Unfortunately they are not applicable in the weighted case.

Aichholzer and Aurenhammer [1] and Palfrader et al. [7] discuss a more practical algorithm. Their algorithm, based on a kinetic triangulation, computes the straight skeleton of a planar straight line graph (PSLG) in $\mathcal{O}(n^3 \log n)$ time. The main idea is to maintain a triangulation of the interior of the wavefront over time. By analyzing the triangles of this kinetic triangulation one can find the event points where the wavefront changes. The cubic worst-case time bound stems from the number of *flip events* (when a reflex wavefront vertex crosses a diagonal of the triangulation).

Huber and Held [6] introduce an $\mathcal{O}(n^2 \log n)$ time and linear space approach to compute the straight skeleton of a given PSLG. Flip events are avoided by utilizing the motorcycle graph induced by the input. We extend their work to positively weighted straight skeletons over simple polygons without holes. Our adaptation of their algorithm leads to an $\mathcal{O}(n^2 + r^3/k + (nr + r^2) \log n)$ time and $\mathcal{O}(n + kr)$ space complexity for a fixed $k \in \mathbb{Z}^+$ such that $k < r$. A space-time tradeoff on k allows to scale the required space from linear to $\mathcal{O}(n + r^2)$ thereby scaling the k dependent term between r^2 and r^3 time. These variants yield practical candidates for implementation. In the sequel we review the algorithm by Huber and Held [6] and explain the modifications required to make their algorithm applicable for positive edge weights.

2 Algorithm

For the triangulation-based algorithm convex input may result in $\Omega(n^2 \log n)$ running time. Nevertheless, the best known upper bound for this approach still is $\mathcal{O}(n^3 \log n)$. Huber and Held [6] show that for every simple polygon there exists a Steiner triangulation with $\mathcal{O}(n)$ Steiner points that is free of flip events. In the weighted case additional Steiner points are needed since the faces of $\mathcal{S}(P, \sigma)$ do not have to be monotone. The total number of Steiner points needed is still in $\mathcal{O}(n)$ and it can be shown that Theorem 1 holds.

Theorem 1 *Every simple polygon P with n vertices and positive edge weights admits a triangulation with $\mathcal{O}(n)$ Steiner points that is free of flip events during the wavefront propagation.*

In the triangulation-based approach the flip events are caused by reflex vertices crashing into diagonals of the triangulation. In [6] this is prevented by employing a Steiner triangulation. To apply this idea we resort to the motorcycle graph induced by P. A motorcycle m starts at every reflex vertex v of $\mathcal{W}_P(0, \sigma)$. Its position at time t is given by $v(t)$. Furthermore, the boundary of P is seen as a solid *wall*. Thus a motorcycle not only crashes at a trace of another motorcycle but as well at the boundary of P. We denote the motorcycle graph of the unweighted P by $\mathcal{M}(P)$, and the weighted motorcycle graph induced by a weighted P by $\mathcal{M}(P, \sigma)$.

The algorithm by Huber and Held [6] relies on two main properties of $\mathcal{M}(P)$: (i) All reflex arcs of $\mathcal{S}(P)$ have to be covered by segments of the motorcycle graph, and (ii) $\mathcal{M}(P)$ induces a convex tessellation of P.

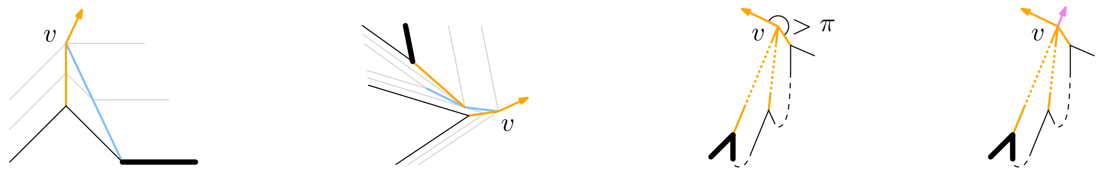
By adding additional motorcycles with different starting times, Huber and Held show that both (i) and (ii) hold for their induced (unweighted) motorcycle graph over any PSLG (even if multi-split events occur).

In the weighted approach, however, both properties (i) and (ii) are violated: Property (ii) does not hold as $\mathcal{M}(P, \sigma)$ does not induce a convex tessellation of P, cf. Fig. 1c. Furthermore, an edge event involving a reflex vertex may result in another reflex wavefront vertex v, cf. Fig. 1a. The arc traced out by v is not part of $\mathcal{M}(P, \sigma)$ as v is not a reflex vertex of $\mathcal{W}_P(0, \sigma)$, thus violating property (i). Updating $\mathcal{M}(P, \sigma)$ at times of such events is expensive, as redirecting one motorcycle may result in recomputing all other motorcycles. Since such *reflex-preserving* edge events may occur after split events, cf. Fig. 1b, also tracking the initial convex arcs with additional motorcycles is insufficient. Clearly $\Omega(n)$ reflex-preserving edge events can occur. Hence, $\mathcal{M}(P, \sigma)$ seems unsuitable for covering all reflex arcs of $\mathcal{S}(P, \sigma)$.

2.1 Extended Wavefront Propagation

We therefore take a different approach and define the arrangement $\mathcal{A}(P)$ induced by the reflex vertices of $\mathcal{W}_P(0, \sigma)$. For every reflex vertex v of $\mathcal{W}_P(0, \sigma)$ a line segment is added to $\mathcal{A}(P)$ such that it starts at v, lies on the ray defined by $v(t)$, and ends at the point where it first connects to a vertex or edge of P, for $t > 0$. $\mathcal{A}(P)$ consists of r such *segments* and covers all reflex arcs of the initial wavefront.

Lemma 2 *$\mathcal{A}(P)$ induces a convex tessellation of P.*



(a) (b) (c) (d)

Figure 1: In (a-b), we see a reflex-preserving edge event at v. In (d-e), we see a multi-split event at v. The input is drawn in black (thick for higher edge weight). In (c-d), convex arcs are drawn in blue and reflex arcs in orange.

Theorem 1 tells us that there always exists a flip-event-free Steiner triangulation. Rather than using the straight-skeleton based Steiner triangulation we use the arrangement $\mathcal{A}(P)$ to track the reflex arcs of $\mathcal{S}(P,\sigma)$. And instead of using a kinetic triangulation we employ an *extended wavefront* $\mathcal{W}_P^*$ to trace out $\mathcal{S}(P,\sigma)$ without flip events:

Definition 3 (Huber and Held [6]) *The overlay of $\mathcal{W}_P(t,\sigma)$ and $\mathcal{A}(P) \cap \bigcup_{t' \geq t} \mathcal{W}_P(t',\sigma)$ defines the extended wavefront $\mathcal{W}_P^*(t,\sigma)$.*

$\mathcal{W}_P^*(t,\sigma)$ is seen as a kinetic PSLG where the vertices which are not in $\mathcal{W}_P(t,\sigma)$ are called *Steiner vertices*. Furthermore, Steiner vertices that belong to both $\mathcal{W}_P(t,\sigma)$ and $\mathcal{A}(P)$ are called *moving Steiner vertices*, while those Steiner vertices which have not yet been reached by the wavefront are called *resting Steiner vertices*, cf. Fig. 2. For each segment s in $\mathcal{A}(P)$ we store four vertices: $v(t)$ of $\mathcal{W}_P(t,\sigma)$, its moving Steiner vertex $v(t')$ where s ends, and the two intersection points on s in $\mathcal{A}(P)$ closest to $v(t)$ and $v(t')$, which are both resting Steiner vertices. Due to multi-split and reflex-preserving edge events *updates* of $\mathcal{A}(P)$ are required during the propagation of $\mathcal{W}_P^*(t,\sigma)$. An update consists of inserting and removing a segment from $\mathcal{A}(P)$. First $\mathcal{W}_P^*(0,\sigma)$ is determined. Then for every edge its collapse time (if finite) is inserted as an event into a priority queue $\mathcal{Q}$ sorted by event time. As $\mathcal{Q}$ is sequentially dequeued the following events are distinguished, which are equivalent to the unweighted scenario:

Edge event: Two vertices u and v meet; the respective straight skeleton arcs are added; u and v are merged into a new vertex w. If w is reflex, i.e., resembles a reflex-preserving edge event, then a new segment on $w(t)$ is added to $\mathcal{A}(P)$; the previous one is removed. Additionally it is checked whether u and v cause a whole triangle of the wavefront to vanish.

Split event: If a reflex vertex u meets a moving Steiner vertex v then the reflex straight skeleton arc traced out by u is added. Consider the wavefront to the left of the edge $e = \overline{uv}$. If this side collapsed then the corresponding straight skeleton arcs are added. Otherwise a new convex vertex emerges, which is connected to the vertices adjacent to u and v lying left to e. Similarly on the right side of e.

Switch event: A convex vertex u meets a moving Steiner vertex or a reflex vertex v. Then u migrates from one convex face to a neighboring one by jumping over v. If v is reflex it becomes a moving Steiner vertex; respective straight skeleton arcs are added.

Multi split event: Reflex vertices $u_0,\dots,u_{k-1}$ meet simultaneously at a resting Steiner vertex u. We number them clockwise around u. First, reflex straight skeleton arcs are added for $u_0,\dots,u_{k-1}$ and their corresponding segments are removed from $\mathcal{A}(P)$. Second, for all consecutive pairs $u_i, u_{1+i \pmod k}$, with $0 \leq i < k$: Let e_i denote the edge $\overline{uu_i}$ and let e_{i+1} denote the edge $\overline{uu_{1+i \pmod k}}$. Then the wavefront is patched for each sector bound by e_i and e_{i+1} as follows. (Note that if $k=0$ then one sector spans the whole local disc.) A new vertex v is created which patches the ccw-edge e_l of e_i at u_i and the cw-edge e_r of e_{i+1} at $u_{1+i \pmod k}$ together. Also note that additional edges e may have been incident to u between e_i and e_{i+1}. Such an edge e could lie exactly on the trajectory of v, e.g., if v is a reflex wavefront vertex, because e_l and e_r span a reflex angle. In this case e, which was incident to u, simply becomes incident to v. Also we add a segment to $\mathcal{A}(P)$ that lies on $v(t)$. As v is reflex, the edge $\overline{uv}$ splits the non-convex sector into two sectors. If one of them is non-convex we add another segment to $\mathcal{A}(P)$ that lies on either $u_1(t)$ or $u_k(t)$ and starts at u, such that the non-convex sector is split into two convex sectors, cf. Fig. 1c and 1d. The next intersection point of these segments is added as resting Steiner vertex to $\mathcal{W}_P^*(t,\sigma)$ and the corresponding edges.

In all other cases where v is convex, e splits e_l resp. e_r by an additional moving Steiner vertex, depending on whether e lies left or right to the trajectory of v.

Start event: A moving Steiner vertex u that moves on segment s of $\mathcal{A}(P)$ meets a Steiner vertex v. This is similar to a multi split event with $k=0$, except that u is not a reflex vertex. Thus, no straight skeleton arc is traced out by u. The next intersection point v' on s in $\mathcal{A}(P)$ is added as resting Steiner vertex to $\mathcal{W}_P^*(t,\sigma)$ as well as the edge $\overline{vv'}$. We also shorten the segments of $\mathcal{A}(P)$ incident at v where no moving Steiner vertex reached v: Their endpoint is modified to v and forms a moving Steiner vertex of $\mathcal{W}_P^*(t,\sigma)$.

When two moving Steiner vertices meet they can be removed. Other events are guaranteed not to occur.

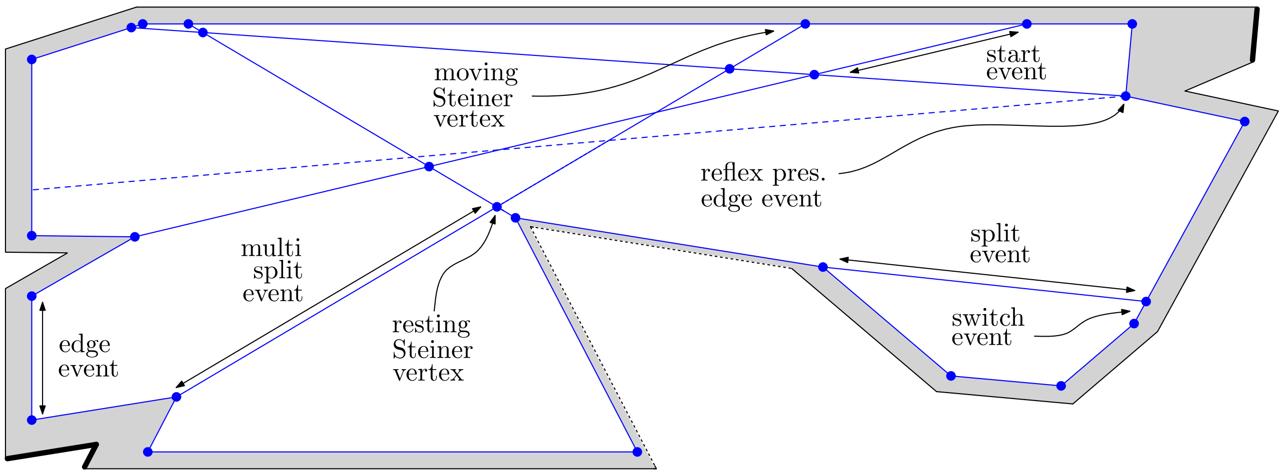


Figure 2: All edges of P have unit weight except those marked in thick (dotted) black, which have large (small, resp.) edge weights; $\mathcal{W}_P^*(t,\sigma)$ is drawn in blue, the blue dashed edge marks a segment removed from $\mathcal{A}(P)$ due to a reflex-preserving edge event.

When $\mathcal{Q}$ is empty also the last component of $\mathcal{W}_P^*(t,\sigma)$ has vanished and $\mathcal{S}(P,\sigma)$ is complete. During the extended wavefront propagation $\mathcal{A}(P)$ is adapted such that the following Lemma 4 holds. Lemma 5 follows from Lemmas 2 and 4.

Lemma 4 *The movement of the reflex vertices of $\mathcal{W}_P(t,\sigma)$ is tracked by $\mathcal{A}(P)$ at any time $t \geq 0$.*

Lemma 5 *For any $t \geq 0$ the set $P \setminus \bigcup_{t' \in [0,t]} \mathcal{W}_P^*(t,\sigma)$ consists of open convex faces.*

2.2 Runtime Analysis

The initial extended wavefront $\mathcal{W}_P^*(0,\sigma)$ and the initialization of $\mathcal{Q}$ can be done in $\mathcal{O}(n \log n + nr)$ time. We have $\mathcal{O}(nr)$ switch events and $\mathcal{O}(r^2)$ start events while all other events occur $\mathcal{O}(n)$ times. Handling one (reflex preserving) edge event takes $\mathcal{O}(n+r+r\log n)$ time; including updating $\mathcal{A}(P)$ in $\mathcal{O}(r)$ time and adding/removing a segment of the wavefront in $\mathcal{O}(n)$ time. The latter may invalidate $\mathcal{O}(r)$ events in $\mathcal{Q}$ as the new segment can intersect $\mathcal{O}(r)$ segments in $\mathcal{A}(P)$ closer to $\mathcal{W}_P(t,\sigma)$ than the current event points. Each requires a $\mathcal{O}(\log n)$ time queue operation. One start event takes $\mathcal{O}(r + \log n)$ time. Where $\mathcal{O}(r)$ time is needed to find the next intersection in $\mathcal{A}(P)$ and $\mathcal{O}(\log n)$ time to add the event to $\mathcal{Q}$. Overall we need $\mathcal{O}(nr + n(n+r+r\log n) + r^2(r+\log n))$ time within linear space to compute $\mathcal{S}(P,\sigma)$. Since $r \in \mathcal{O}(n)$, this simplifies to $\mathcal{O}(n^2 + nr\log n + r^3)$ time.

Assume that we can afford $\mathcal{O}(n+kr)$ space, for a fixed k with $1 \leq k \leq r$. Let s of $\mathcal{A}(P)$ be a segment which we query for the next intersection point p. Instead of computing just p in $\mathcal{O}(r)$ time we compute and store the next k intersection points in $\mathcal{O}(r \log n)$ time. Note that this pre-computed data can be invalidated by at most $\mathcal{O}(n)$ edge events. Since every segment has $\mathcal{O}(r)$ intersection points we have to compute the next k intersections only every $\mathcal{O}(r/k)$ times. This allows to deduce the following Theorem 6.

Theorem 6 *This algorithm computes $\mathcal{S}(P,\sigma)$ of P in time $\mathcal{O}(n^2 + r^3/k + nr\log n)$ time and $\mathcal{O}(n+kr)$ space.*

References

[1] O. Aichholzer and F. Aurenhammer. Straight Skeletons for General Polygonal Figures in the Plane. In *Proc. 2nd COCOON*, volume 1090 of *LNCS*, 1996.

[2] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A Novel Type of Skeleton for Polygons. *J. Univ. Comp. Sci*, 1(12):752–761, 1995.

[3] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader. Weighted Straight Skeletons in the Plane. *CGTA*, 48(2):120–133, 2015.

[4] S.-W. Cheng, L. Mencel, and A. Vigneron. A Faster Algorithm for Computing Straight Skeletons. *ACM Trans. Algorithms*, 12(3):44:1–44:21, Apr. 2016.

[5] D. Eppstein and J. Erickson. Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions. *Discrete Comput. Geom.*, 22(4):569–592, 1999.

[6] S. Huber and M. Held. A Fast Straight-Skeleton Algorithm Based on Generalized Motorcycle Graphs. *IJCGA*, 22(5):471–498, 2012.

[7] P. Palfrader, M. Held, and S. Huber. On Computing Straight Skeletons by Means of Kinetic Triangulations. In *ESA'12*, 2012.

[8] A. Vigneron and L. Yan. A Faster Algorithm for Computing Motorcycle Graphs. *Discrete Comput. Geom.*, 52(3):492–514, Oct. 2014.