International Journal of Computational Geometry
& Applications
Vol. 29, No. 3 (2019) 251–267

DOI: 10.1142/S0218195919500080



# Recognizing Geometric Trees as Positively Weighted Straight Skeletons and Reconstructing Their Input

Günther Eder*, Martin Held† and Peter Palfrader‡
*FB Computerwissenschaften, University of Salzburg*
*5020 Salzburg, Austria*
**geder@cs.sbg.ac.at*
†*held@cs.sbg.ac.at*
‡*palfrader@cs.sbg.ac.at*

Received 25 April 2019
Accepted 11 September 2019
Published 25 October 2019
Communicated by S.-W. Cheng

We extend results by Biedl *et al.* (ISVD'13) on the recognition and reconstruction of straight skeletons: Given a geometric tree $G$, can we recognize whether $G$ resembles a weighted straight skeleton $\mathcal{S}$ and, if so, can we reconstruct an appropriate polygonal input $P$ and an appropriate positive weight function $\sigma$ such that $\mathcal{S}(P, \sigma) = G$? We show that a solution polygon $P$ and a weight function $\sigma$ can be found in $O(n)$ time and space for a geometric tree $G$ with $n$ faces if at most one node of $G$ has two incident edges that span an angle greater than $\pi$. In addition, we show that $G$ implicitly encodes enough information such that all other weighted bisectors of any solution $P$ can be obtained from $G$ without explicitly computing $P$.

*Keywords*: Weighted straight skeleton; bisector graph; recognition.

## 1. Introduction

A planar straight-line graph (PSLG) is an embedding of a planar graph such that all its edges are mapped to straight-line segments which only intersect at common endpoints. As in Ref. 4, we use the term PSLG$^\infty$ to denote a "*planar straight-line graph with infinity*" by allowing straight-line rays in addition to straight-line segments as edges. Still, no two edges intersect except at common endpoints. It is convenient to add a vertex at infinity as second endpoint for rays to ensure that



†Corresponding author.

every edge is defined by two vertices. All conventional vertices of a PSLG$^\infty$ are called *finite* vertices. To distinguish the two sides of an edge of a PSLG$^\infty$, we assign a direction to every edge, thus obtaining a *left* and a *right* side.

In this paper, we consider a recognition/reconstruction problem, where we seek a solution $H$ for a given input PSLG$^\infty$, $G$, in the plane $\mathbb{R}^2$. In a nutshell, $H$ may form a simple polygon or even a PSLG$^\infty$ itself, and $G$ shall represent its (weighted) straight skeleton. To distinguish elements of $H$ and $G$ we call edges of $G$ *arcs* and vertices of $G$ *nodes*. As usual, a *face* induced by $G$ is a maximal open and connected region in $\mathbb{R}^2$ that does not contain an arc or ray of $G$.

Wavefront propagation is a standard means for defining and computing straight skeletons[1,5,11] of polygons and PSLGs. It models an offsetting process where, for a given simple polygon $P$, all edges move inwards in a self-parallel manner and at unit speed. The wavefront edges of two adjacent polygon edges $e_1, e_2$ are joined by a wavefront vertex that moves along the angular bisector of $e_1$ and $e_2$ such that the resulting polygon remains simple. At time $t$ of the wavefront propagation this polygon is called the *wavefront*, $P_t$, at time $t$. We have $P_0 := P$. For sufficiently small $t > 0$ the wavefront $P_t$ will remain combinatorially equivalent to $P$. Otherwise events need to be handled in order to maintain the simplicity of $P_t$: We get an edge event if an edge shrinks to zero length and vanishes, a split event if an edge is split into two, or a multi event if multiple such events occur simultaneously at the same position. (A vertex event[9] is an example of a multi event.) The wavefront propagation stops once all components of $P_t$ have vanished. Then the straight skeleton $\mathcal{S}(P)$ of $P$ is given by the traces of all vertices of $P_t$ over the time of the propagation. It is easy to see that all arcs of $\mathcal{S}(P)$ lie on angular bisectors of edges of $P$ and, thus, are formed by straight-line segments. The same wavefront propagation can be applied to the exterior of $P$, thus obtaining the exterior straight skeleton of $P$. A further extension allows to define straight skeletons of PSLGs. Note, though, that both straight skeletons will contain (straight-line) rays as arcs. We use the term $P_t^+$ to explicitly denote an inwards wavefront propagation of a polygon $P$, and $P_t^-$ for an outwards propagation.

*Weighted straight skeletons* are obtained by relaxing the requirement that all input edges move at unit speed. Now the input edges move with speeds induced by edge weights, with one (multiplicative) weight per edge. (In theory, we could also use two different weights for the two sides of an input edge.) Unweighted and weighted straight skeleton have many applications; see, e.g.,[5,10] and the references contained therein. In recent years a variety of algorithms were introduced for computing (weighted) straight skeletons of simple polygons with and without holes as well as of PSLGs.[2,5–9,11]

## 2. Our Contribution

The question "Given a geometric graph $G$, can we reconstruct an appropriate polygonal input $H$ such that $\mathcal{S}(H) = G$?", was asked by Biedl *et al.*[4] Aichholzer

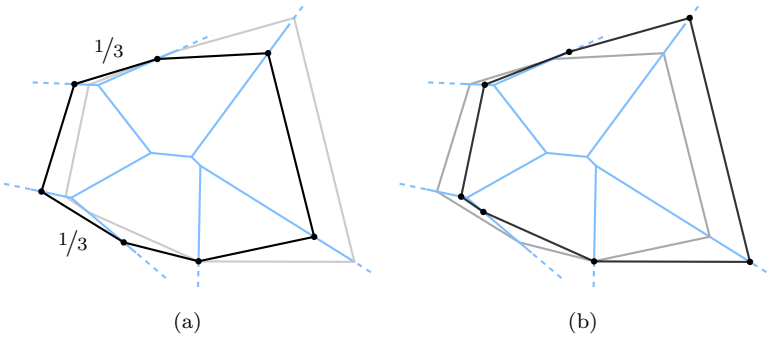


Fig. 1. (Color online) For a given input $G$ (blue) we see two solution polygons (black and gray).

*et al.*[3] discuss under which conditions a directed tree is a combinatorial representation of a straight skeleton as well as how to construct an input polygon. We extend the work by Biedl *et al.*[4] to weighted straight skeletons by allowing (multiplicative) edge weights in $H$. For instance, for the blue PSLG$^\infty$ shown in Fig. 1(a), the convex black polygon with uniform edge weights except for two edges that have weight $1/3$ constitutes a feasible solution for our problem. (That is, the weighted straight skeleton of the black convex polygon is the blue graph.) A second solution with different weights is illustrated by the black polygon shown in Fig. 1(b).

While our primary focus lies on reconstructing a simple polygon $H$ as input, we will also allow $H$ to be a PSLG$^\infty$ where all finite vertices are of degree at least two. Let $\sigma$ be a weight function which provides a strictly positive edge weight $\sigma(e)$ for every edge $e$ of $H$. We denote by $\mathcal{S}(H, \sigma)$ the weighted straight skeleton of $H$ relative to the weight function $\sigma$.

After reviewing a characterization of weighted straight skeletons, we study different classes of input graphs and solve Problems 1 and 2.

**Problem 1.** Let $G$ be a PSLG$^\infty$ which is a geometric tree $G$. Can we find a PSLG$^\infty$ $H$ and a weight function $\sigma$ such that $\mathcal{S}(H, \sigma) = G$?

**Problem 2.** Let $G$ be a PSLG$^\infty$ which is a geometric tree $G$. Can we determine the bisectors between non-adjacent faces of $G$ without actually identifying $H$?

## 3. Preliminaries

Consider the supporting lines $\ell_i, \ell_j$ of two distinct edges of a simple polygon $P$, with weights and interior sides inherited from their defining edges. Then $\ell_i, \ell_j$ define a unique bisector line: We propagate both $\ell_i$ and $\ell_j$ to their common interior (exterior) side with speeds that correspond to their weights. Then their bisector is traced out by the intersection of these two moving lines. In case two distinct lines $\ell_i, \ell_j$ are parallel, we follow the procedure of the wavefront propagation and propagate both

lines only in their interior side: In case they coincide on a line $\ell$ after finite time, $\ell$ is their bisector.[a]

Let $H_{t,\sigma}$ denote the weighted wavefront of the PSLG$^\infty$ $H$ at time $t$ relative to the weight function $\sigma$. Likewise, we denote a wavefront fragment stemming from an edge $e$ of $H$ by $e_t$. Any such fragment will be contained in one of the two offset supporting lines $\ell(e) \pm n_e \cdot \sigma(e) \cdot t$, where $\ell(e)$ is the supporting line of $e$ and $n_e$ is a unit normal vector. (Hence, the larger the weight $\sigma(e)$ the faster the edge fragments $e_t$ move during the wavefront propagation.) Recall that a wavefront vertex $v_t$ of $H_{t,\sigma}$ moves on the bisector defined by its two incident wavefront edges. Note that the $n$ supporting lines, one for each edge of $H$, form an *inducing line set* for the bisector graph and the straight skeleton traced out by $H_{t,\sigma}$.

Biedl *et al.*[5] show that many properties of unweighted straight skeletons are preserved for positively weighted straight skeletons of simple polygons $P$. In particular $\mathcal{S}(P,\sigma)$ is connected, is a tree, and has no crossings. It consists of $n+v-1$ arcs, where $v$ denotes the number of straight skeleton nodes and $n$ the number of vertices of the simple polygon $P$. Every face $f$ of $\mathcal{S}(P,\sigma)$ is guaranteed to be a weakly simple polygon.[b] Furthermore, $\mathrm{cl}(P_{t+\varepsilon,\sigma}) \subsetneq \mathrm{cl}(P_{t,\sigma})$ for any $\varepsilon > 0$, where cl() denotes the closure of the area bounded by the polygon(s). These properties generalize to a PSLG$^\infty$ $H$ and $\mathcal{S}(H,\sigma)$ in a natural way, provided that $H$ does not have a finite node of degree one.

Biedl *et al.*[4] state three necessary properties for a graph $G$ to match $\mathcal{S}(P)$ which we adapt for the weighted setting. The following properties are necessary for a simple polygon $P$ and its weighted straight skeleton $\mathcal{S}(P,\sigma)$: (i) All nodes of $\mathcal{S}(P,\sigma)$ are of degree at least three. (ii) Each face $f$ of $\mathcal{S}(P,\sigma)$ contains exactly one segment of $P$. (iii) Every edge of $P$ begins and ends on an arc of $\mathcal{S}(P,\sigma)$.

## 4. $G$ is a Star Graph

We start with assuming that our input PSLG$^\infty$ $G$ is a *star graph*. That is, $G$ consists of a finite node $v$ and $n$ rays, $b_1, \ldots, b_n$, that originate at $v$ and are numbered in counter-clockwise order (CCW) around $v$. In this section we consider different types of graphs $H$ such that $\mathcal{S}(H,\sigma)$ matches $G$. For a start we assume that the rays of $G$ induce a convex partition of the plane: Every pair of consecutive rays of $G$ bounds a convex wedge.

### 4.1. *H is a star-shaped polygon*

We construct a star-shaped polygon $P$ with $n$ vertices in the following way: For $1 \le i \le n$, we choose an arbitrary point $p_i$ in the interior of every ray $b_i$. (That is,

[a] Additional ambiguities may occur when parallel lines are present; details and best-practice methods are described by Biedl *et al.*[5]

[b] A polygon is 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.

the point $p_i$ may not coincide with $v$.) Then we connect every consecutive pair $(p_i, p_{i+1})$ of points by a line segment $e_i$. Of course, we wrap around the indices and let $p_0 := p_n$ and $p_{n+1} := p_1$.

How can we define an appropriate weight function $\sigma$ such that $\mathcal{S}(P, \sigma)$ equals $G$? Let $d_\perp(v, e_i)$ denote the normal distance between $v$ and the supporting line $\ell(e_i)$ of $e_i$. Note that $e_i$ will have moved a distance of $\sigma(e_i) \cdot t$ from $e_i$ towards $v$ after time $t$. Hence, setting

$$\sigma(e_i) := d_\perp(v, e_i) \tag{1}$$

ensures that all wavefront edges of $P_{t_v,\sigma}$ within the interior of $P$ will reach $v$ at time $t_v := 1$.

It remains to argue that the vertices of $P_{t,\sigma}$ do indeed move along $G$. We know that a weighted straight skeleton has straight-line edges. That is, all vertices of $P_{t,\sigma}$ move along straight lines. Recall that the vertices of $P_{0,\sigma}$ coincide with the vertices of $P$. Furthermore, due to the setting of Eq. (1), the offset supporting line of edge $e_i$ reaches $v$ at time $t = 1$. This implies that the offset supporting lines of two neighboring edges $e_i$ and $e_{i+1}$ of $P$ intersect at $v$ at time $t = 1$, thus fixing the second point of the line that contains the weighted bisector between $e_i$ and $e_{i+1}$. We conclude that the vertex of $P_{t,\sigma}$ that corresponds to $p_i$ moves along the line through $p_i$ and $v$, that is, along $b_i$. When propagating $P_{t,\sigma}$ to its exterior the vertices stay on the rays of $G$ as the weight of an edge is applied to both sides and thus the respective bisector for two edges is equivalent for both sides.

Besides the solutions that can be obtained by offsetting $P$ inwards or outwards we can move the points $p_i$ of $P$ along the rays of $G$ arbitrarily, thus creating a vast family of solutions $P$ which, assuming the weights are computed according to Eq. (1), satisfy $\mathcal{S}(P, \sigma) = G$. In particular, $P$ need not be convex. See Fig. 2(a).

Note that the correctness of this approach hinges upon the fact that $v$ lies within the kernel of $P$, which is implied by our construction. This observation leads to the following additional result: Suppose that we are given a star-shaped polygon $P$. We can choose an arbitrary point within (the interior of) the kernel of $P$ and declare it to be our node $v$. If we set the weights of the edges of $P$ as defined in Eq. (1) then the resulting weighted straight skeleton $\mathcal{S}(P, \sigma)$ will always be a star graph rooted at $v$.

### 4.2. *H is a star graph*

We continue with investigating other types of solutions. If one vertex of $H$ coincides with $v$ and, thus, two (or more) edges meet at $v$, then all edges of $H$ have to meet at $v$. (An edge whose supporting line does not contain $v$ would have to have an infinite weight.) Hence, in this scenario $H$ has to be a PSLG$^\infty$ that forms a star graph. Since every face of $G$ has to contain one edge of $H$, we conclude that $H$ consists of $n$ rays which originate at $v$ and which need to be placed such that the rays of $G$ and $H$ alternate in the cyclic order around $v$.

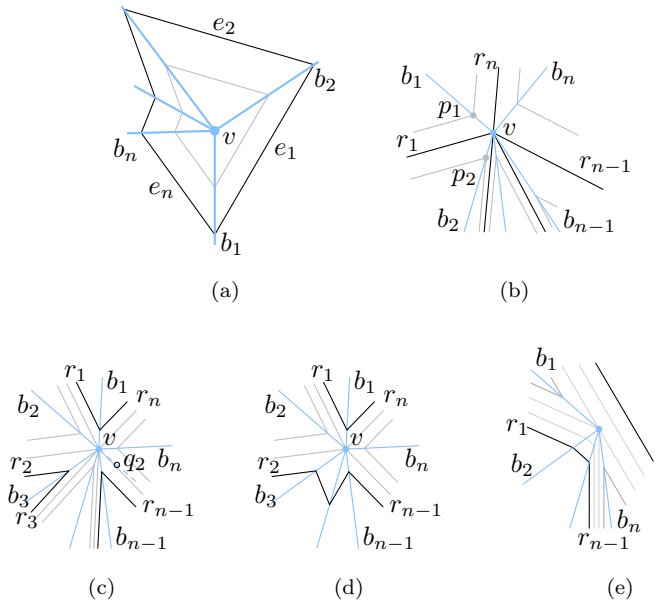


Fig. 2. (Color online) In (a) we show a polygon $H$ (black) such that $\mathcal{S}(H, \sigma)$ matches the star graph (blue). In (b) we see a solution that is itself a star graph; distinct wavefront offsets (gray). In (c)–(e) further solutions are drawn where $H$ is a disconnected PSLG$^\infty$.

Suppose that the rays $\{r_1, \ldots, r_n\}$ of $H$ are numbered in CCW order around $v$, with ray $r_i$ being placed between $b_i$ and $b_{i+1}$. The first $n-1$ rays $\{r_1, \ldots, r_{n-1}\}$ may be placed arbitrarily within the respective faces of $G$. Let the weight of the first ray be fixed: $\sigma(r_1) := 1$. We construct the wavefront offsets on both sides of $r_1$ for time $t := 1$. Let $p_1$ and $p_2$ be the intersections of these offsets with $b_1$ and $b_2$. Then $\sigma(r_2) := d_\perp(p_2, r_2)$. We apply this approach consecutively around $v$ until we know $\sigma(r_{n-1})$ and $p_n$. To obtain $r_n$ we look at the midpoint $q$ of the line segment $\overline{p_1 p_n}$. Now let the ray $r_n$ go through $q$. The intercept theorem implies that the normal distances $d_\perp(p_n, r_n)$ and $d_\perp(p_1, r_n)$ of $p_n$ and $p_1$ to $r_n$ are identical, and we can set $\sigma(r_n) := d_\perp(p_n, r_n)$.

A sample star graph can be seen in Fig. 2(b). Note that this approach works also if one of the angles $\angle(b_i, b_{i+1})$ is greater than or equal to $\pi$, provided that the ray $r_i$, that lies in the non-convex face $f_i$, partitions $f_i$ into two convex faces.

### 4.3. *H is a disconnected PSLG$^\infty$*

Additional solutions $H$ can be determined if we allow $H$ to form a disconnected PSLG$^\infty$. If the number of rays $n$ in $G$ is even then we may form a multi event at $v$ such that $n/2$ reflex wavefront vertices of $P_t$ meet at $v$ at the same time $t_v$. As illustrated in Fig. 2(c), we place a reflex vertex $p_i$ arbitrarily at every other ray $b_i$ of $G$. Every such vertex $p_i$ forms the start of a v-shaped component of $H$ whose geometry is determined as follows: Starting with $i := 1$, we place a ray $r_i$ (with

start point $p_i$) arbitrarily within the wedge formed by $b_i$ and $b_{i+1}$, where $r_i$ is not parallel to $b_{i+1}$. The supporting line of $r_i$ intersects the supporting line of $b_{i+1}$ in a point $q_{i+1}$ outside of $b_{i+1}$. The ray $r_{i+1}$ starts at $p_{i+2}$ and is contained in the line through $p_{i+2}$ and $q_{i+1}$. Our construction ensures that $r_{i+1}$ lies within the wedge defined by $b_{i+1}$ and $b_{i+2}$. We then increment $i$ by 2 and repeat this process until all $n$ rays of our $n/2$ v-shaped components have been constructed.

The weight of $r_i$ is obtained as $\sigma(r_i) := d_\perp(v, r_i)$. This setting ensures that all wavefronts which propagate towards $v$ will reach $v$ at time $t_v := 1$. Thus, the wavefront vertices that stem from $p_i$ travel along $b_i$ towards $v$ or towards infinity. Since, for every odd $i$, the rays $r_i$ and $r_{i+1}$ intersect on the supporting line of $b_{i+1}$, it is guaranteed that the convex wavefront vertex formed by weighted offsets of $r_i$ and $r_{i+1}$ will travel along $b_{i+1}$ for time $t > 1$. This set-up can be modified by replacing a pair of rays $r_i$ and $r_{i+1}$, for some odd $i$, by two finite straight-line segments from $p_i$ to a new vertex $u_{i+1}$ and from $u_{i+1}$ to $p_{i+2}$, cf. Fig. 2(d). The new vertex $u_{i+1}$ can be chosen arbitrarily along $b_{i+1}$, provided that the appropriate edge weights are applied to $\overline{p_i u_{i+1}}$ and $\overline{u_{i+1} p_{i+2}}$.

If $n$ is odd then we pursue a similar approach: Starting with $b_1$ we place reflex vertices $p_i$ on every other ray of $G$. We construct rays $r_1, r_2, \ldots, r_{n-1}$ as outlined above. Finally, we add the line segment $\overline{p_1 p_n}$ to $H$. The weights of all rays and of the finite line segment are chosen as above.

Both scenarios can be adapted if a non-convex partition of the plane is induced by the rays of $G$. In the sample setting shown in Fig. 2(e), the infinite line passes through the intersection of the supporting line of $r_1$ with the supporting line of $b_1$, and through the intersection of the supporting line of $r_{n-1}$ with the supporting line of $b_n$.

## 5. $G$ is a Tree

Let $G$ form a geometric tree whose arcs do not intersect except at common nodes and whose finite nodes all have degree at least three. We regard it as a PSLG$^\infty$ which partitions the plane into $n$ unbounded faces. The leaves of $G$ form rays that we assume to meet in a node at infinity. Ofcourse, these leaf rays may not intersect except at common nodes. The following lemma is easily derived from Euler's polyhedron formula.

**Lemma 1.** *Let $G$ be a geometric tree that partitions the plane into $n \geq 3$ unbounded faces. Then $G$ contains at most $2n - 3$ arcs and at most $n - 2$ finite nodes.*

We start with assuming that every finite node $v$ of $G$ is of degree three. Note that the bounds given in Lemma 1 are sharp in this case. Clearly, every node $v$ is the result of an event during a propagation of $P_t$. Since $v$ is of degree three, it can have been formed only by an edge event or by a split event. Now consider a sufficiently small disk $D$ centered at $v$. If the three arcs incident at $v$ induce a sector

of $D$ that is not convex then a reflex vertex of $P$ was involved in this event. This observation motivates Lemma 2.

**Lemma 2.** *Let every finite node $v$ of $G$ be of degree three and let the three edges incident at every $v$ induce a convex partition of the plane around $v$. Then $\mathcal{S}(P,\sigma) = G$ can hold for a polygon $P$ and a strictly positive weight function $\sigma$ only if $P$ is convex.*

**Proof.** Let $\sigma$ be a strictly positive weight function. Since every node $v$ in $G$ is of degree three, it is created by an edge or split event. We start by showing that every node $v$ of $\mathcal{S}(P,\sigma)$ induces a local convex partition if $P$ is convex: The results of[5] tell us that $\mathcal{S}(P,\sigma)$ is connected, is a tree, and has no crossings. The convexity of $P$ implies that only edge events can occur during the interior wavefront propagation. Due to the strictly positive edge weights every edge event produces locally a convex partition on the respective node. A solution polygon $P$ has one vertex on every ray of $G$. (Otherwise $P$ cannot have an edge in every face of $G$ as $G$ is a tree.) Thus, the exterior wavefront propagation of $P_{t,\sigma}$ contains no finite event and all vertices of $G$ induce locally a convex partition.

We continue with the second direction. Assume that $\mathcal{S}(P,\sigma) = G$ holds for a polygon $P$. If every node $v$ of $G$ is of degree three and induces locally a convex partition then $P$ is convex: Assume to the contrary that $P$ has at least one reflex vertex $v_r$: Then, a vertex of $P_{t,\sigma}$ starts at $v_r$ and traces at least one *reflex arc* of $G$. Let every arc $a$ of $G$ inherit the direction defined by its respective moving wavefront vertex of $P_{t,\sigma}$. We define a directed path $p$ that contains all reflex arcs that can be reached with one directed walk from $v_r$. Let $a'$ form the last arc in $p$. We look at the node $v$ where $a'$ ends.

If a split event took place at $v$ then we arrive at a contradiction as every split event induces a non-convex partition locally around the respective node: The split event is formed at $v$ where the wavefront edges $e'_t$ and $e''_t$ that trace out $a'$ meet an opposing wavefront edge $e_t$. Then, due to the split event, $e_t$ is split into $e^l_t$ and $e^r_t$. As illustrated in Fig. 3(a), the wavefront around $v$ defines two wedge-like untraced areas $\wedge_1$ and $\wedge_2$, one bordered by $e^l_t$ and $e'_t$, and the second one bordered by $e^r_t$ and $e''_t$. The two arcs that start at $v$ each lie in one of $\wedge_1$ and $\wedge_2$. Hence, the respective new bisectors form a non-convex wedge at $v$, cf. Fig. 3(a).

If $v$ was formed by an edge event then let $e_l$, $e_m$, and $e_r$ denote the three wavefront edges at $v$ where $e_m$ is the edge that vanishes. Two arcs are incoming at $v$ and span a convex wedge $\wedge$, illustrated as green area in Fig. 3(b). The vertex $v$ can only induce a convex partition if the supporting line $\ell(b)$ of the outgoing arc $b$ at $v$ lies in $\wedge$. By definition, $\ell(b)$ goes through the intersection $v'$ of the supporting lines of $e_l$ and $e_r$ as well as through $v$. If the vertex between $e_l$ and $e_m$ is reflex (convex, resp.), then the vertex between $e_m$ and $e_r$ is convex (reflex, resp.). In either case, $v'$ cannot lie in $\wedge$ and, thus, $v$ cannot induce locally a convex partition.

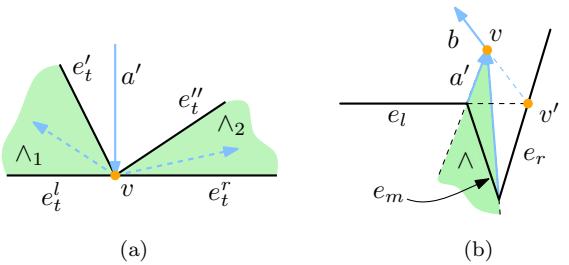


Fig. 3. In (a) we see a split event and in (b) we see an edge event at $v$. In both (a), (b) a non-convex partition is induced by the edges incident at $v$.

Thus, we arrive at a contradiction to our assumption and conclude that $P$ has to be convex. □

### 5.1. *G Induces a convex partition*

Assume that the edges at every node of $G$ induces a convex partition of the plane. We seek a polygon $P$ such that $\mathcal{S}(P, \sigma) = G$. Lemma 2 implies that every solution $P$ is necessarily convex. To construct $P$, we start by selecting an arbitrary finite node $v_r$ of $G$ and choose it as the root of $G$, thus turning $G$ into a rooted tree. We form a triangle $\Delta$ by choosing one point arbitrarily within the relative interior of each of the three arcs incident at $v_r$. As in Sec. 4, the weights of the edges of $\Delta$ are fixed such that $v_r$ is reached at time $t := 1$.

Next, we use $\Delta$ as our initial convex polygon and apply a *reverse wavefront propagation*: We start with $Q^-_{0,\sigma} := \Delta$ and propagate the wavefront $Q^-_{t,\sigma}$ outwards as time increases. A *reverse* edge event occurs when $Q^-_{t,\sigma}$ reaches a finite node $v$ of $G$ at time $t_v$. Let the edges $e'_t$ and $e''_t$ of $Q^-_{t,\sigma}$ be incident at $v$ at this time, and let $a_l, a_r, a_m$ (left, right, middle) be the arcs of $G$ incident at $v$. The common endpoint of $e'_t$ and $e''_t$ has moved along $a_m$ during time $t_v - \varepsilon$, for some small $\varepsilon > 0$. As illustrated in Fig. 5(a), we create a new edge $e_t$ between $e'_t$ and $e''_t$. (At time $t = t_v$ this new edge will have length zero.) At time $t := t_v + \varepsilon$, for some small $\varepsilon > 0$, one endpoint of $e'_t$ will move along $a_l$ and one endpoint of $e''_t$ will move along $a_r$, cf. Fig. 5(b). Hence, also the movement of $e_t$, and thus its direction and weight, is uniquely determined.

Since the arcs of $G$ induce a convex partition of the plane around each node, a reverse edge event replaces one convex vertex of $Q^-_{t,\sigma}$ with two convex vertices: Let $v_t$ be the intersection of $e'_t$ and $e''_t$ over time, and let $Q^*_t$ be the polygon $Q^-_{t,\sigma}$ but without the changes affected by the reverse edge event. The vertex $v_t$ of $Q^*_t$ has moved along $a_m$ prior to $t_v$ and continues into the wedge spanned by $a_l$ and $a_r$ after $t_v$. Furthermore, after the event time, the edge $e_t$ (which has as its endpoints the intersections of $e'_t$ and $e''_t$ with $a_l$ and $a_r$) is contained in $Q^*_t$, connecting two (adjacent) edges of $Q^*_t$, thus creating a shortcut in its boundary. Hence, $Q^-_{t,\sigma}$ is the intersection of $Q^*_t$ with a half-plane bounded by the supporting line of $e_t$, and

$Q^-_{t,\sigma}$ remains convex under reverse edge events. We apply this reverse wavefront propagation until the last finite node of $G$ has been processed at time $t_e$. After this time, $Q^-_{t,\sigma}$ will not encounter any further event. We now let $P := Q^-_{t^\star,\sigma}$, where $t^\star := t_e + \varepsilon$ for some arbitrary $\varepsilon > 0$, and let $P$ inherit the weight function $\sigma$. From this construction the following lemma follows directly:

**Lemma 3.** *The polygon $P$ obtained from a reverse wavefront propagation starting at an arbitrary finite node of $G$ is convex. The (standard) wavefront propagation of $P$ under the weight function obtained from the reverse propagation will yield $G$ as the weighted straight skeleton of $P$, i.e., $\mathcal{S}(P,\sigma) = G$.*

Let $\mathcal{H}$ be the family of all convex polygons $P$ (and weight functions $\sigma$), with $\mathcal{S}(P,\sigma) = G$, such that the interior wavefront propagation of $P$ finally results in a single wavefront triangle that collapses. We can obtain every member of $\mathcal{H}$ by choosing some finite node $v$ of $G$ as root node $v_r$ and by choosing the vertices of the initial triangle $\Delta$ within the relative interiors of the arcs of $G$ incident to $v$. All remaining solutions $P$ where the final collapsing wavefront $P^+_t$ is formed by a quadrilateral can be obtained by a similar process.

It is easy to see that we can abolish our assumption that all finite nodes are of degree exactly three and still obtain solutions: If the initial root node $v_r$ has degree $d > 3$, then we generate an arbitrary convex $d$-gon as our initial polygon $Q$. If $Q^-_{t,\sigma}$ encounters a node of degree $d$ of $G$, then we add $d-2$ new edges, where $d-3$ new vertices have some additional freedom. All we need to do in either case is to select the new vertices such that $Q^-_{t,\sigma}$ remains convex at all times.

We note, however, that our approach need not find all solutions if nodes of higher degree are present. Let the node $v$ of $G$ be of degree four and the incident arcs at $v$ locally induce a convex partition around $v$. Clearly, $v$ can be formed by an event where two incoming arcs are traced by reflex wavefront vertices. That is, a degree-four node of $G$ may result in non-convex polygons becoming feasible solutions.

### 5.2. *G Does not induce a convex partition*

We now allow one node $v_0$ of $G$ such that the three edges around $v_0$ do not locally induce a convex partition of the plane. How can we find a solution polygon $P$ such that $\mathcal{S}(P,\sigma) = G$?

Every solution $P$ has to have one vertex on every ray of $G$. (Otherwise $P$ cannot fulfill the necessary condition to have exactly one edge in every face of $G$, since $G$ is a tree.) This observation implies Lemma 4, otherwise a solution $P$ could not form a simple polygon. It also tells us that $v_0$ cannot be the only finite node of $G$. (Otherwise, $P$ cannot be a simple polygon since the rays at $v_0$ would not induce a convex partition.) The proof of Lemma 5 follows from Lemma 2.

**Lemma 4.** *Let $P$ be a simple polygon and $G$ be a PSLG$^\infty$ which forms a geometric tree. Consider a translation of every ray of $G$ such that its finite node coincides with*

*some fixed point $p$. If $\mathcal{S}(P, \sigma) = G$ holds for some strictly positive weight function $\sigma$ then the rays incident at $p$ induce a convex partition of the plane, i.e., every wedge hinged at $p$ has an angle strictly less than $\pi$. Furthermore, the cyclic order of the rays around $p$ is the same as the order in which they are intersected by any solution polygon $P$.*

**Lemma 5.** *Let $a_l, a_m, a_r$ (left, middle, and right) be the three arcs incident at $v_0$ in CCW order such that $a_l$ and $a_r$ define a non-convex sector. Then at least one arc has to be traced out by a reflex wavefront vertex.*

During the wavefront propagation $v_0$ was produced due to a split event or an edge event. We analyze the directions in which the vertices of the wavefront move in a neighborhood of $v_0$: Due to Lemma 5 at least one reflex vertex ends at $v_0$: Three incoming wavefront vertices where one is reflex are not possible since a wavefront triangle cannot have a reflex vertex. An outgoing reflex wavefront vertex would result in a second node of $G$ whose arcs do not induce a convex partition; see Fig. 4(c). Hence, three scenarios remain: (i) both $a_l$ and $a_r$ are outgoing and convex, Fig. 4(a); (ii) $a_l$ is incoming and $a_r$ is outgoing where both are convex Fig. 4(b); and (iii) $a_r$ is incoming and $a_l$ is outgoing where both are convex.

Hence, $a_m$ is formed by a ray of $G$ that has to contain a reflex vertex of $P$. If an edge event occurred at $v_0$ then a convex wavefront vertex moved along $a_l$ towards $v_0$ and then continued along $a_r$, or the other way around $a_r$, $v_0$, and $a_l$. If a split event occurred then a (reflex) wavefront vertex moved along $a_m$ towards $v_0$, and then two (convex) wavefront vertices continued along $a_l$ and $a_r$.

We are now ready to construct a suitable polygon $P$. Let $f$ be the face of $G$ that has $a_l$ and $a_r$ on its boundary and, thus, an angle greater than $\pi$ at $v_0$. Due to Lemma 4 we can choose a line $\ell$ that intersects both rays contained in the boundary of $f$ such that $v_0$ and all other finite nodes of $G$ lie on the same side of $\ell$. (The line $\ell$ can be chosen arbitrarily except for these two conditions.) Our final polygon $P$ will have an edge which is parallel to $\ell$ and whose two endpoints lie on these two rays. The node $v_0$ splits the boundary of $f$ into two branches, a left branch that contains $a_l$ and a right branch that contains $a_r$. Within the left branch we determine the finite node $v_l$ whose normal distance from $\ell$ is maximal among all finite nodes of the left branch. Same for $v_r$ within the right branch. Since $G$ is guaranteed to have at least two finite nodes, at least one of $v_l$ or $v_r$ has to exist. W.l.o.g., at least the left branch has finite nodes and $v_l$ exists.

We construct a small triangle $\Delta$ around $v_l$ by placing three vertices on the three arcs of $G$ that are incident at $v_l$. The placement is arbitrary except for the condition

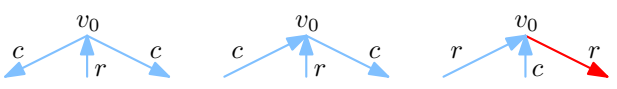


Fig. 4. Arrangements of directed edges such that a non-convex partition is induced at $v_0$.

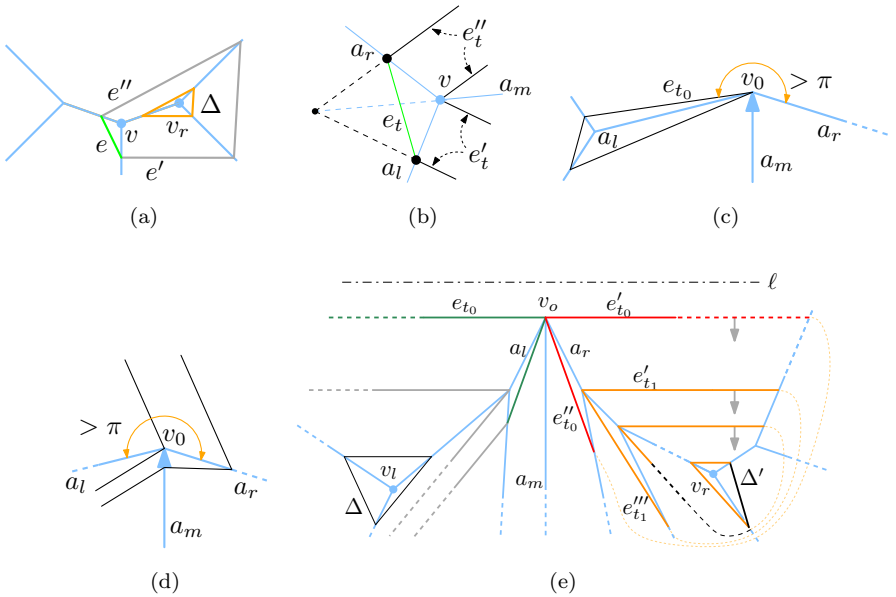


Fig. 5. (Color online) (a) A tree $G$ (blue) with an initial polygon $\Delta$. An edge (green) is added in a reverse edge event. (b) Details of a reverse edge event. (c)–(e) Incoming and outgoing arcs at a non-convex node $v_r$; the vertical arc depicts $a_m$. (c), (e) A reverse split event, showing how the second wavefront is formed using $\Delta'$ in (e).

that the edge $e$ of $\Delta$ that lies within $f$ has to be parallel to $\ell$. Again we choose the weights of the edges of $\Delta$ such that $v_l$ is reached at time $t := 1$. We use $\Delta$ as our initial polygon $Q$ and apply a reverse wavefront propagation $Q^-_{t,\sigma}$, with $Q^-_{0,\sigma} := \Delta$. This wavefront propagation proceeds by applying reverse edge events until $v_0$ is reached for some specific time $t_0$. Let $e_{t_0}$ be the edge of $Q^-_{t_0,\sigma}$ that is parallel to $e$ and therefore also parallel to $\ell$. If the CCW angle between $a_r$ and $e_{t_0}$ is smaller than $\pi$ then we treat $v_0$ as a reverse edge event, cf. Fig. 5(d). All remaining finite nodes of $G$ correspond also to reverse edge events, and we proceed with our wavefront propagation until all finite nodes have been processed. Otherwise $v_0$ resembles a *reverse split event*, cf. Fig. 5(c). Note that in this case the right branch of the boundary of $f$ has at least one finite vertex $v_r$. Otherwise, $\ell$ could not intersect an arc in the right branch of the boundary of $f$. We explain how to establish a triangle $\Delta'$ around $v_r$, cf. Fig. 5(e). As in the case of $\Delta$, all three vertices of $\Delta'$ will lie on the three arcs of $G$ that are incident at $v_r$, and the edge of $\Delta'$ that lies within $f$ has to be parallel to $\ell$, thus imposing a constraint on two vertices of $\Delta$. However, we do not have the freedom to choose the third vertex of $\Delta'$ arbitrarily. Note that the wavefront of our yet unknown polygon $P$ will consist of one non-convex polygon if $v_0$ has not yet been reached, and will split up into two convex polygons at the time when $v_0$ is reached. One of these polygons, $Q^-_{t_0,\sigma}$, contains $e_{t_0}$ while the second polygon contains $e'_{t_0}$ and a second edge $e''_{t_0}$ which also has $v_0$ as endpoint. This edge

$e''_{t_0}$ lies in the face of $G$ which is bounded by $a_m$ and $a_r$; its geometry and weight is uniquely determined. (Recall that $e_{t_0}$ and $e'_{t_0}$ are derived from the same edge of $P$, whose weight is given by the weight of $e$ defined when setting up $\Delta$.)

Let us construct this second polygon. We know only the geometry and weight of two of its edges, $e'_{t_0}$ and $e''_{t_0}$. But let us pretend that we would know all its edges and their weights. A wavefront propagation will eventually lead to $\Delta'$ and then reach $v_r$. If this wavefront propagation allows offsets of $e'_{t_0}$ and $e''_{t_0}$ to reach $v_0$ then two edges of $\Delta'$ are known, thus fixing $\Delta'$. (The weight of the third edge of $\Delta'$ is chosen such that all three edges of $\Delta'$ reach $v_0$ at the same time.) Otherwise, the only second possible case is that $e''_{t_0}$ vanishes in an edge event at some time $t_1$. In this case a new edge $e'''_{t_1}$ will become the CCW neighbor of $e'_{t_1}$ in the current wavefront at time $t_1$. As for $e''_{t_0}$, the geometry and weight of $e'''_{t_1}$ is uniquely determined. We now focus on $e'_{t_1}$ and $e'''_{t_1}$ and proceed with the wavefront propagation. Eventually we are guaranteed to know two edges of $\Delta'$, thus fixing $\Delta'$ also in this case.

As soon as both the geometry and the edge weights for $\Delta$ and $\Delta'$ are known we can start reverse wavefront propagations from both triangles. Suppose that an offset edge of $e'_{t_0}$ reaches $\Delta'$ at time $t^\star$. If $t_0 \geq t^\star$ then we start an reverse wavefront propagation at $\Delta$ at time 0 and at $\Delta'$ at time $t_0 - t^\star$. Otherwise we start at $\Delta'$ at time 0 and at $\Delta$ at time $t^\star - t_0$. During both wavefront propagations we process reverse edge events until both wavefronts will reach $v_0$ at time $\max\{t_0, t_\star\}$. To handle the reverse split event we unite the two wavefronts. Then we proceed with processing reverse edge events until all finite vertices of $G$ have been encountered.

This construction gives us a family of polygons $P$ and weights $\sigma$ such that $\mathcal{S}(P, \sigma) = G$ holds. Again, we can waive the constraint on the degrees of the finite nodes and allow arbitrary degrees greater than three for all nodes except for $v_0$.

In Theorem 1 we summarize the time and space complexity required to form a solution polygon for Problem 1.

**Theorem 1.** *Let $G$ be a PSLG$^\infty$ which forms a geometric tree whose arcs and rays do not intersect except at common nodes. Let $n$ be the number of unbounded faces induced by $G$. If $G$ has at most one node such that the edges around that node do not locally induce a convex partition then we can find a solution polygon $P$ with $\mathcal{S}(P, \sigma) = G$ in $\mathcal{O}(n)$ time and space.*

**Proof.** Selecting an arbitrary finite vertex $v$ of $G$ and forming the initial triangle $\Delta$ around $v$ takes $\mathcal{O}(1)$ time. Processing one reverse edge event during the reverse wavefront propagation takes constant time. In total there are $\mathcal{O}(n)$ many nodes and, thus, also $\mathcal{O}(n)$ many events. These nodes can be traversed in, e.g., depth-first order without the need for a priority queue because subtrees and their events are independent from each other.

A reverse split event at vertex $v_r$ stops one reverse wavefront propagation, finds an appropriate new root vertex, and starts another reverse wavefront propagation. In our scenarios, we choose the new root $v'$ in $\mathcal{O}(1)$ time, as it is a neighbor of $v_r$.

Clearly the second reverse wavefront propagation processes different vertices of $G$ than the first one, except for $v_r$. Therefore $P$ can be obtained in $\mathcal{O}(n)$ time and space. □

### 5.3. *Implicit bisectors of G*

Let $G$ be a geometric tree such that every node of $G$ is of degree three, and assume that there exists a solution polygon $P$ and weight function $\sigma$ such that $\mathcal{S}(P, \sigma) = G$. (We do not require these to be known, though.) Recall that a solution polygon $P$ and a weight function $\sigma$ define $\binom{n}{2}$ bisectors, one for each pair of edges of $P$. In this section we introduce a method to construct all bisectors that are not part of $G$, but are implicitly defined by $P$ and $\sigma$. Our approach is guaranteed to work if all (implicitly defined or explicitly given) bisectors are pairwise non-parallel. Therefore, assume that $P$ is without parallel bisectors.

An arc $a_{ij}$ of $G$ is on the boundary of exactly two faces, $f_i$ and $f_j$, traced out in a propagation process by polygon edges $e_i$ and $e_j$, and this arc is contained in the weighted bisector $b_{i,j}$ between these polygon edges. As this bisector line $b_{i,j}$ is the same for all solutions $(P, \sigma)$, we may consider it independent of any specific solution and refer to it as the *bisector between faces* $f_i$ *and* $f_j$. Similarly, we say that a node of $G$ is *equidistant to three faces* (relative to the edge weights of $P$). For technical reasons we assume that any triple of bisectors intersects in a unique point. In particular we do not allow four bisectors to intersect in a common point.

Let us analyze a small tree $G$ with only two finite vertices. Then $G$ has one finite arc, four rays, and thus defines four faces, as illustrated in Fig. 6. Let $f_1$ and $f_3$ form faces of $G$ that share no common arc in $G$. We can construct (the supporting line of) the bisector $b_{1,3}$ between $f_1$ and $f_3$, i.e., the bisector of the non-adjacent edges of $P$ that traced out $f_1$ and $f_3$, by extending existing arcs of $G$: The intersection $p_{1,3,4}$ is formed by intersecting $b_{1,4}$ and $b_{3,4}$, and $p_{1,2,3}$ is formed by intersecting $b_{1,2}$ and $b_{2,3}$.

By construction, $p_{1,3,4}$ is equidistant to the faces $f_1$, $f_3$, and $f_4$, relative to the edge weights of $P$ implied by the arcs of $G$, and $p_{1,2,3}$ is equidistant to the faces $f_1$, $f_2$, and $f_3$. Thus, $b_{1,3}$ can be formed by the line through $p_{1,3,4}$ and $p_{1,2,3}$.

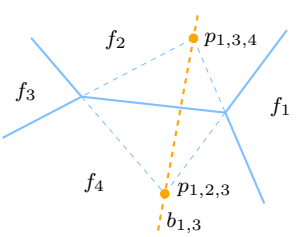


Fig. 6. Computing the bisector between two faces that share neighbors.

In this canonical setting, $f_1$ and $f_3$ border the common faces $f_2$ and $f_4$ such that it was obvious which arcs to intersect to obtain $b_{1,3}$. To construct bisectors for two faces $f_i, f_m$ in a larger tree $G$ where they do not share common neighbors, we first have to compute one or more intermediate bisectors between other faces. We now show how to identify and obtain those bisectors.

We define the dual graph $\mathcal{D}$ of $G$ such that every vertex $f'$ of $\mathcal{D}$ corresponds to a face $f$ of $G$, and two vertices of $\mathcal{D}$ are connected by an edge if the respective faces in $G$ share a common arc and, thus, we know the bisector between them. Since $G$ consists only of unbounded faces, $\mathcal{D}$ is an outerplanar graph. The edges that bound the outer face of $\mathcal{D}$ correspond to the unbounded rays of $G$, and all other edges ("*diagonals*") map to arcs between finite nodes of $G$. Furthermore, since all nodes of $G$ are of degree three, all interior faces of $\mathcal{D}$ are triangles. Hence, $\mathcal{D}$ is a biconnected maximal outerplanar graph, and we can regard (the planar embedding of) $\mathcal{D}$ as a convex $n$-vertex polygon which is triangulated, where $n$ is the number of faces of $G$.

Every triangle $\Delta$ in $\mathcal{D}$ corresponds to a node in $G$, i.e., the location where three bisectors intersect. We can modify $\mathcal{D}$ and $G$ by carrying out edge flips in $\mathcal{D}$ and mirroring the corresponding changes in $G$, cf. Fig. 7.

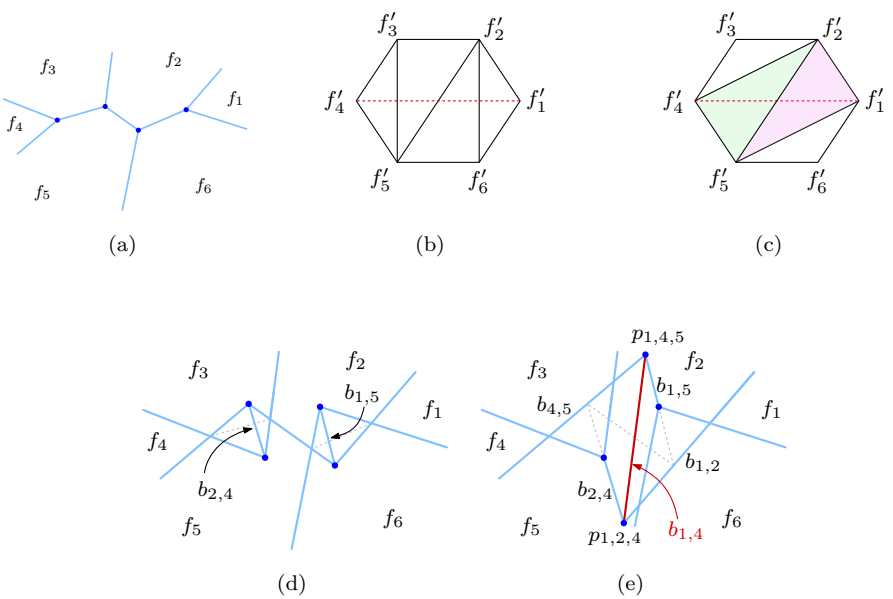


Fig. 7. (a) Given a geometric tree $G$ that is the weighted straight skeleton of a polygon, we want to construct the bisector induced by the polygon edges that traced out faces $f_1$ and $f_4$. This bisector does not depend on the specific polygon. (b) The dual graph of $G$ lets us learn which edges to flip to obtain the triangulation edge $\overline{f'_1 f'_4}$ and, thus, the bisector $b_{1,4}$. First we flip $\overline{f'_3 f'_5}$ and $\overline{f'_2 f'_6}$, resulting in the dual seen in (c) and the primal graph in (d). Lastly, we flip $\overline{f'_2 f'_5}$, thereby constructing $b_{1,4}$ in the primal (e).

**Lemma 6.** *Let $d$ be a diagonal in $\mathcal{D}$ such that $d = \overline{f'_j f'_n}$ and let $\Delta_1(f'_i, f'_j, f'_n)$ and $\Delta_2(f'_m, f'_n, f'_j)$ be the two triangles which share $d$. Then the edge flip on $d$ in $\mathcal{D}$ produces the new bisector line $b_{i,m}$ in $G$.*

**Proof.** Recall that all edges of $\Delta_1$ and $\Delta_2$ correspond to arcs in $G$, i.e., bisector portions. Thus, $\Delta_1$ gives us the bisectors $b_{i,j}$ and $b_{i,n}$, and $\Delta_2$ gives us $b_{m,n}$ and $b_{m,j}$. To construct $b_{i,m}$, we need two points where $f_i$ and $f_m$ are equidistant. The bisector $b_{i,n}$ intersects $b_{m,n}$ at the point $p_{i,m,n}$. Similarly, $b_{i,j}$ intersects $b_{j,m}$ at $p_{i,j,m}$. Then $b_{i,m}$ can be formed by the line through $p_{i,m,n}$ and $p_{i,j,m}$. Note that these intersections are guaranteed to exist since we had assumed that no two bisectors are parallel to each other.

In $G$, we now remove the arc on $b_{j,n}$, which is common to $\Delta_1$ and $\Delta_2$. Then we extend or shorten the arcs on the four other bisectors of the triangles involved to be incident at $p_{i,m,n}$ and $p_{i,j,m}$ (in a new pairing), and we introduce a new arc between $p_{i,m,n}$ and $p_{i,j,m}$. By flipping $d$ also in the triangulation $\mathcal{D}$ to connect $f'_i$ and $f'_m$, we complete the edge flip. □

Since $\mathcal{D}$ forms a triangulation it contains exactly $n-3$ diagonals. To obtain a certain diagonal $d = \overline{f'_i f'_j}$ that is not in $\mathcal{D}$, we apply repeated edge flipping: We identify all diagonals that cross $d$ and flip them. The last diagonal which crossed $d$ will become $d$ when flipped.

A single edge flip in $\mathcal{D}$ can be carried out in constant time as described in Lemma 6. Furthermore, we can identify all edges that need to be flipped in amortized linear time, and each edge is flipped at most once. Thus, we derive Theorem 2, thereby solving Problem 2.

**Theorem 2.** *Let $G$ be a PSLG$^\infty$ which forms a geometric tree such that every node of $G$ is of degree three, and let $n$ be the number of unbounded faces induced by $G$. For any two faces $f_i$ and $f_j$ of $G$ we can construct the unique bisector $b_{i,j}$ in $\mathcal{O}(n)$ time.*

**Corollary 1.** *The $2n-3$ arcs of $G$ implicitly define all $\binom{n}{2}$ bisectors.*

Note that these bisectors are independent of any specific polygon $P$ and weight function $\sigma$ with $\mathcal{S}(P, \sigma) = G$. In particular, we can obtain the bisectors from $G$ without first determining a polygon and weight function.

## 6. Conclusion and Discussion

In this work, we gain further insight into the structure of weighted straight skeletons as well as general bisector graphs. In particular, a PSLG$^\infty$ $G$ which forms a geometric tree such that every node of $G$ is of degree three provides already sufficient information to deduce all bisectors: If we assume that there exists a solution polygon $P$ and a weight function $\sigma$ such that $\mathcal{S}(P, \sigma) = G$ then the bisector between

any pair of edges of $P$ can be obtained from $G$ without a need to know $P$ and $\sigma$ themselves. This is surprising because, in general, an infinite number of polygons $P$ (and corresponding weight functions $\sigma$) will serve as solutions. Still, all bisectors between pairs of edges of $P$ are fixed by $G$.

If at most one of the faces induced by $G$ contains a node that is not convex then we can also reconstruct polygonal input $H$ and a weight function $\sigma$ such that $\mathcal{S}(H, \sigma) = G$. Some lengthy technical details would allow to extend this result slightly to one face having several nodes that are not convex. But allowing two faces with one non-convex node each seems to derail a successful reconstruction completely. Similarly, the presence of just one bounded face complicates matters drastically, even if all faces are convex. We hope that our results will sparkle new work on this problem which will finally lead to a solution for an unrestricted PSLG$^\infty$.



## Acknowledgments

Work supported by Austrian Science Fund (FWF) Grant P25816-N15 and ORD 53-VO.

## References

1. O. Aichholzer and F. Aurenhammer, Straight skeletons for general polygonal figures in the plane, in *Proceedings of the Annual International Conference on Computing and Combinatorics* (1996).
2. O. Aichholzer *et al.*, A novel type of skeleton for polygons, *J. Universal Computer Science* **1**(12) (1995) 752–761.
3. O. Aichholzer *et al.*, Representing directed trees as straight skeletons, in *Proceedings of the International Symposium on Graph Drawing and Network Visualization* (2015), pp. 302–313.
4. T. Biedl, M. Held and S. Huber, Recognizing straight skeletons and Voronoi diagrams and reconstructing their input, in *Proceedings of the International Symposium on Voronoi Diagrams in Science and Engineering* (2013), pp. 37–46.
5. T. Biedl *et al.*, Weighted straight skeletons in the plane, *Comput. Geom.*: *Theory Appl.* **48**(2) (2015) 120–133.
6. S.-W. Cheng, L. Mencel and A. Vigneron, A faster algorithm for computing straight skeletons, *ACM Trans. Algorithms* **12**(3) (2016) 44:1–44:21.
7. S.-W. Cheng and A. Vigneron, Motorcycle graphs and straight skeletons, *Algorithmica* **47**(2) (2007) 159–182.
8. G. Eder and M. Held, Computing positively weighted straight skeletons of simple polygons based on bisector arrangement, *Inform. Process. Lett.* **132** (2018) 28–32.
9. D. Eppstein and J. Erickson, Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions, *Discr. Comput. Geom.* **22**(4) (1999) 569–592.
10. M. Held and P. Palfrader, Straight skeletons with additive and multiplicative weights and their application to the algorithmic generation of roofs and terrains, *Computer-Aided Design* **92**(1) (2017) 33–41.
11. S. Huber and M. Held, A fast straight-skeleton algorithm based on generalized motorcycle graphs, *Int. J. Comput. Geom.* **22**(5) (2012) 471–498.