Recognizing Straight Skeletons and Voronoi Diagrams and Reconstructing Their Input

Therese Biedl*
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 1A2, Canada
Email: biedl@uwaterloo.ca

Martin Held
FB Computerwissenschaften
Universität Salzburg
5020 Salzburg, Austria
Email: held@cosy.sbg.ac.at

Stefan Huber
Institute of Science and Technology Austria
3400 Klosterneuburg, Austria
Email: stefan.huber@ist.ac.at

***Abstract*—A straight skeleton is a well-known geometric structure, and several algorithms exist to construct the straight skeleton for a given polygon or planar straight-line graph. In this paper, we ask the reverse question: Given the straight skeleton (in form of a planar straight-line graph, with some rays to infinity), can we reconstruct a planar straight-line graph for which this was the straight skeleton? We show how to reduce this problem to the problem of finding a line that intersects a set of convex polygons. We can find these convex polygons and all such lines in $O(n \log n)$ time in the Real RAM computer model, where n denotes the number of edges of the input graph. We also explain how our approach can be used for recognizing Voronoi diagrams of points, thereby completing a partial solution provided by Ash and Bolker in 1985.**

I. Introduction

The straight skeleton $\mathcal{S}(P)$ of a polygon P is a well-known geometric data structure. It is defined by offsetting a polygon inwards, thereby moving all edges at constant speed, and tracing the movement of the polygon's vertices. The straight skeleton is always a tree for a polygon without holes. For convex polygons the straight skeleton coincides with the Voronoi diagram, but for non-convex polygons the two concepts differ: Voronoi diagrams have parabolic arcs at every reflex vertex, while the straight skeleton consists entirely of straight-line segments.

This procedural definition of a straight skeleton extends naturally to the interior and exterior of a polygon, by computing inwards and outwards offsets. Straight skeletons have also been generalized to arbitrary planar straight-line graphs (PSLGs), with an appropriate special handling of vertices of degree one. In both cases the resulting skeleton forms a planar straight-line graph, with rays to infinity; see Fig. 1. We refer to Huber and Held [1] for an extensive and up-to-date discussion of theory and applications of straight skeletons.

Many algorithms are known for computing the straight skeleton of a polygon or a planar straight-line graph; see [1] and the references therein. In this paper, we consider the reverse question: Given a planar straight-line graph G with rays to infinity, is there another planar straight-line graph H such that the straight skeleton of H is G?

* Supported by NSERC. Research done while visiting Univ. Salzburg.

A. Related results

Many related reconstruction questions have been studied in the literature. One of the older ones is *Delaunay realizability*: Given a planar graph G, is there a set of points such that the dual graph of the Voronoi diagram of the points is G? This has been answered in the affirmative for all 4-connected planar graphs and all outer-planar graphs, cf. [2], [3]. In particular, it follows from this result that every tree is the Voronoi diagram of some set of points in convex positions. (This result was provided independently in [4].) Recently, it was shown that every tree is realizable as the Voronoi diagram of a suitable convex polygon [5]. Recall that for convex polygons the Voronoi diagram coincides with the straight skeleton.

In the above references, the graph was given abstractly, i.e., without fixing positions for the vertices or any other geometric properties. In contrast to this, we are interested in the case where the graph is given as *geometric graph*, i.e., with a fixed drawing in the plane. The corresponding reconstruction question for Voronoi diagrams — "Given a planar straight-line graph with rays to infinity, is it the Voronoi diagram of a set of points?" — was investigated already almost 30 years ago. The first reference here seems to be Ash and Bolker [6], who give a characterization if the graph has only vertices of odd degree. The general reconstruction problem for vertices of odd and even degrees was solved a bit later by Hartvigsen [7] in polynomial time, based on a transformation of the problem to linear programming. Aurenhammer's work [8] on reciprocal figures and projection polyhedra also allows to characterize and recognize Voronoi diagrams. The techniques of [7], [8] can also handle higher-dimensional generalizations.

Aichholzer et al. [5] investigated the realizability of a phylogenetic tree as the straight skeleton of a polygon and answered to the affirmative for caterpillar graphs. A phylogenetic tree is an abstract tree where partial geometric properties are given, i.e., the lengths of the edges and the incidence orders of the edges at vertices are fixed.

B. Our contribution

To our knowledge, the problem of reconstructing the input from a straight skeleton (given as a geometric graph) has not previously been investigated by other authors. We gave preliminary results for reconstructing a polygon from its straight skeleton in [9]. In this paper we present two algorithms for a general solution of this problem, in two slightly different

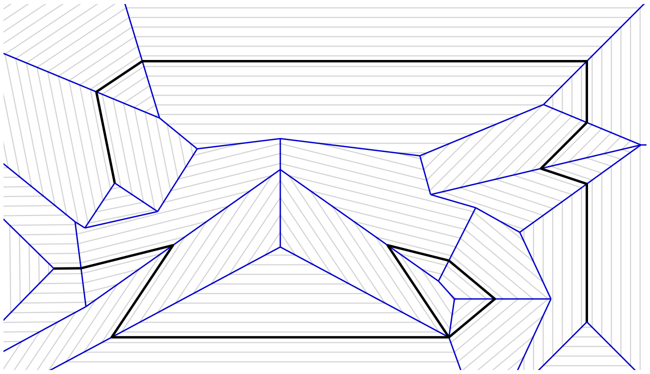

Fig. 1: The straight skeleton (solid) of a PSLG (thick solid); a family of offsets is shown in light gray.

settings. Our approach is similar in spirit to the work by Ash and Bolker [6]. In the general setting (studied in Section IV) we are given a planar straight-line graph G with rays to infinity, and ask whether this could possibly be the straight skeleton of some planar straight-line graph H. We give an algorithm that answers this question in $O(n \log n)$ time, where n is the number of edges of G, and also finds (implicitly) all possible planar straight-line graphs H within a finite subset of the plane.

In order to solve this reconstruction problem, we first (in Section III) need to characterize exactly when a given planar straight-line graph has the form $H \cup \mathcal{S}(H)$, i.e., consists of an input and its straight skeleton. This characterization is not at all trivial, not even in the special case where we know that H is a simple polygon. Aichholzer et al. [10] gave an example of a polygon H and a straight-line tree inside H that satisfies the (obviously necessary) conditions of having edges that are bisectors and having cells that are monotone, yet the tree is not the straight skeleton of H. They left as an open problem of how to characterize when a tree is indeed a straight skeleton of a given polygon; our paper provides such a characterization.

In the last part of the paper (Section V), we consider a partition of the plane into cells, as induced by a PSLG G, and ask whether G can form the Voronoi diagram of a set S of points. We show that our approach employed for recognizing straight skeletons allows to extend the work by Ash and Bolker [6] to general PLSGs without need for imposing a restriction on the degrees of the nodes of G. We give an algorithm that shows the existence in $O(n)$ time in the Real RAM model of computation. For the case that the existence is not unique, we can describe all possible solutions in $O(n \log n)$ time in the Real RAM model of computation.

II. Background

A. Planar straight-line graphs

A *planar straight-line graph* (PSLG) is a geometric graph whose edges are given by a set of n straight-line segments and whose vertices are formed by the endpoints of the line segments. No two line segments of a PSLG intersect except in a common endpoint. We extend this concept to *planar straight-line graphs with infinity* (PSLG$^\infty$) by allowing straight-line rays and straight lines in addition to straight-line segments as edges. Still, all edges do not intersect except at common endpoints. All conventional vertices of this graph are called *finite vertices*. In order to retain the nice property that every edge links two vertices of the graph we apply a one-point compactification of the plane and introduce one *vertex at infinity* which serves as the second endpoint of all rays and as both endpoints of all straight lines.

Then a PSLG$^\infty$ defines the *underlying abstract graph* where two vertices are linked if and only if a segment/ray/line is incident to both of them. (Any line of the PSLG$^\infty$ gives rise to a loop in the graph.) Clearly the graph is planar, and the PSLG$^\infty$ defines the cyclic order of edges around vertices, hence faces of the graph. The faces are in one-to-one correspondence with the *cells* of the PSLG$^\infty$, i.e., the maximal open and connected regions that contain no segment/ray/line. We use graph-theoretic terms (such as edge and face) interchangeably with the geometric counter-part (such as segment/ray/line and cell.)

B. Straight skeletons

The *straight skeleton* $\mathcal{S}(P)$ of a polygon P is a PSLG that describes the movement of vertices while offsetting P inwards. Formally, for a small $t > 0$, let P_t be the polygon obtained by moving all edges inwards in a parallel fashion by distance t, and connecting them in the same order. While t is small enough, this is well-defined and the order of edges does not change. As t gets larger, three or more edges may simultaneously occupy a point in P_t; we call this an *event*. Depending on the configuration of the edges, we either have an *edge event* where one edge of P vanishes, or a *split event* where one edge of P is split into two parts, or multiple such events simultaneously. Then the offset-propagation continues in the resulting polygon(s), until all edges have vanished. The straight skeleton of P consists of all those points that were occupied by a vertex of P_t for some $t > 0$. Many properties of straight skeletons are known. In particular, all bounded cells of the PSLG $\mathcal{S}(P) \cup P$ contain exactly one edge of P and are monotone with respect to that edge. Also, any vertex of $\mathcal{S}(P)$ has degree three or more.

This definition generalizes naturally to the exterior of a polygon (by considering both inward and outward offsetting of the polygon) and to a PSLG H. Some of the straight-skeleton edges then become rays. Vertices of degree one of H must be handled separately: The initial offset at a degree-1 vertex v with incident e to consists of two parallel copies of e, offset by t in either direction, and a third edge $e^\perp$ that is perpendicular to e and has distance t from v. Notice that with this definition the cell of $\mathcal{S}(H)$ defined by offsetting $e^\perp$ does not have a segment of H corresponding to it, and that the straight skeleton has a vertex of degree two at v. Figure 1 shows the straight skeleton of a sample PSLG together with a family of offsets.

C. Problem statement

Problem 1 (GMP-SS). *Given a PSLG$^\infty$ G, can we find a planar straight-line graph H such that $\mathcal{S}(H) = G$?*

We call this problem *GMP-SS*, which stands for "graph-matching problem, using straight skeletons". If we want to refer to a specific instance for a particular input graph G then we write GMP-SS(G). Since any practical application will only be interested in solutions H within a finite subset of the plane, we restrict our search for solutions of GMP-SS(G) to some enlarged copy of the bounding box of all finite vertices of G. A simplified version of GMP-SS considers the case where G is a star graph. That is, we have a single finite vertex v and finitely many rays connecting v with the vertex at infinity. Problem 1 now basically asks whether we can find a polygon H, whose vertices need to lie on the rays emanating from v, such that $\mathcal{S}(H) = G$. This problem is easier to solve, by propagating a solution around v. In fact, we can answer a slightly more general problem where we match an arbitrary geometric tree:

Problem 2 (TMP-SS)**.** *Given a geometric tree G where the leaves are represented by rays to infinity, is there a H such that $\mathcal{S}(H) = G$?*

The acronym TMP-SS stands for "tree-matching problem, using straight skeletons". Crucial to solving GMP-SS will be to characterize when exactly a tree is a straight skeleton of a polygon. Hence as part of our solution we also solve the following problem, first posed by Aichholzer et al. [10].

Problem 3. *Given a polygon P and a geometric tree T inside P whose leaves are at the vertices of P. Give necessary and sufficient conditions for T to be the straight skeleton of P.*

In our preliminary paper [9], we considered only TMP-SS and restricted H to be a convex polygon. The results in this paper extend G from a tree to an arbitrary PSLG$^\infty$ and H from a convex polygon to an arbitrary PSLG.

III. Characterizing a straight skeleton

We now solve Problem 3, i.e., characterizing straight skeletons of polygons, which will be a crucial ingredient for Section IV. We first list three conditions for straight skeletons that are clearly necessary, and then give the (lengthy) proof that they are also sufficient.

For a PSLG H, its straight skeleton $\mathcal{S}(H)$ has the following properties:

1) If a vertex of $\mathcal{S}(H)$ has degree two then it coincides with a degree-one vertex of H. All other vertices of $\mathcal{S}(H)$ have at least degree three.
2) Every face of $H \cup \mathcal{S}(H)$ contains exactly one segment of H, except for faces generated by degree-one vertices of H.
3) Every edge of H begins and ends at an edge of $\mathcal{S}(H)$.

In the remainder of this section we denote by F the set of faces of G. The above properties motivate to denote a solution of Problem 1 as a mapping $\lambda\colon F \to \mathcal{L}$, where $\mathcal{L}$ denotes the set of lines in the plane.

Definition 1 (Inside-condition)**.** *The mapping λ fulfills the* inside-condition *if for all faces f, $\lambda(f)$ intersects f in a single line segment $\sigma(f) := \lambda(f) \cap f$.*

The line segment $\sigma(f)$ may be degenerated into a single point v, but only if v is a vertex of degree one of H. For mappings λ that fulfill the inside-condition, we can construct a graph H whose edges are given by the segments $\sigma(f)$, with $f \in F$. In other words, if $\mathcal{S}(H)$ is indeed G then the offset within f at time t consists of one or more segments that are parallel to $\lambda(f)$ and have distance t. (In case that $\lambda(f) \cap f$ is a single point at a degree-one vertex v, the offset that sweeps f is parallel to $\lambda(f)$.) In the following we denote by G^* the graph $G \cup H$.

We can reformulate Problem 1 to the question whether for a PSLG$^\infty$ G a mapping λ exists such that $\mathcal{S}(H) = G$ holds for the resulting graph H. It is well known that each face f of $\mathcal{S}(H) \cup H$ is monotone with respect to $\lambda(f)$. That is, the boundary of f consists of two monotone polygonal chains. The chain that contains $\sigma(f)$ is called the *lower chain* of f, the other chain is called the *upper chain* of f. (In case that f is unbounded, the upper chain may contain the infinite vertex.) We will reuse this notation for faces of G^* accordingly.

Definition 2 (Sweeping-condition)**.** *A face f of G^* fulfills the* sweeping-condition *if the following properties hold:*

(i) *The face f is monotone.*

(ii) *The lower chain of f is split by $\sigma(f)$ into two monotone sub-chains and the orthogonal distances of each sub-chain's vertices to $\lambda(f)$ are strictly increasing as we consecutively enumerate them starting at $\sigma(f)$.*

The mapping λ fulfills the sweeping-condition *if all faces of G^* fulfill the sweeping-condition.*

A face of $\mathcal{S}(H) \cup H$ fulfills the sweeping-condition. Actually, the lower chain of a straight-skeleton face possesses an even stronger property: it is convex, see [11]. However, this stronger property will later follow from our characterization. For the matter of convenience, we will sometimes phrase property (ii) of the sweeping-condition as "the lower chain of face f has no local maximum except at the ends, and no local minimum except at $\sigma(f)$."

Definition 3 (Bisector-condition)**.** *An edge e of G fulfills the* bisector-condition *if e lies on the bisector of $\lambda(f)$ and $\lambda(f')$, where f, f' denote the two incident faces of e. The mapping λ fulfills the* bisector-condition *if all edges of G fulfill the bisector-condition.*

It is well known that any solution λ to Problem 1 fulfills the inside-, sweeping- and bisector-condition, see, e.g., [11]:

Lemma 4. *Let G be a PSLG$^\infty$, for which all finite vertices have a degree of at least three. If the mapping λ is a solution to Problem 1, then λ fulfills the inside-, sweeping- and bisector-condition.*

The remainder of this section is devoted to proving the converse of Lemma 4, i.e., that the inside-, sweeping-, and bisector condition are also sufficient of λ being a solution to Problem 1. In the following we will assume that λ indeed fulfills these three conditions.

We consider an offsetting process of H with respect to G. Observe that $\lambda(f)$ tessellates $f \in F$ into two faces except for f being the face at a degree-one vertex of H. Let F^* denote the face set of G^*. Hence, a face of F contains one or two faces of F^*. If $f^* \in F^*$ is one of the faces of $f \in F$, then we denote by $\lambda^*_t(f^*)$ the line parallel to $\lambda(f)$ that is in the same half-plane as f^* with respect to $\lambda(f)$. We will reuse the notation $\lambda(f')$ and $\sigma(f')$ also for f and write $\lambda(f)$ and $\sigma(f)$.

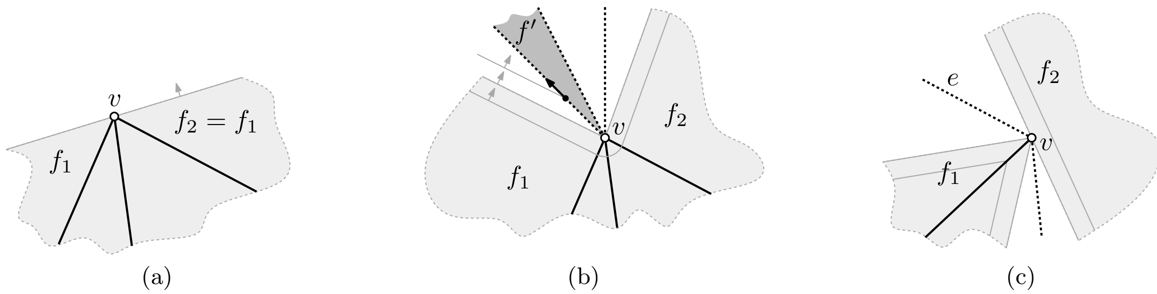

Fig. 2: The local neighborhood of v has been partially swept by either wavefront. The shaded area depicts the swept loci. The thick solid edges show the ingoing edges at v and the thick dotted edges depict the outgoing edges at v. The gray lines show the wavefront at t resp. $t - \epsilon$ with ϵ considered small. (a) It cannot happen that a non-shaded sector contains no outgoing edge. (b) It cannot happen that a non-shaded sector contains more than one outgoing edge as v would be a local minimum in the lower chain of an incident face f'. This sub-figure also depicts a typical edge event where the wavefront within two faces f_1 and f_2 join as the wavefront edges between collapsed. (c) A typical split event happened at v. The wavefront within the face f_2 is split into two edges.

Definition 5. *The* wavefront of H w.r.t. G*, denoted by* $\mathcal{W}_G(t)$*, is defined by*

$$\mathcal{W}_G(t) := \bigcup_{f \in F^*} \lambda_t^*(f) \cap f, \tag{1}$$

where $t \geq 0$ *denotes the time. By* $\mathcal{W}_{\mathcal{S}(H)}(t)$ *we denote the ordinary straight-skeleton offset of* H *at time* t*.*

The sweeping-condition says that for increasing t, we can illustrate $\mathcal{W}_G(t) \cap f$, with $f \in F^*$, as a sweeping process of f. That is, $\mathcal{W}_G(t) \cap f$ starts as a single segment and is split at every local minimum[1] of the upper chain of f. (If two or more local minima are connected by a sequence of edges parallel to $\lambda(f)$ then the segment is only split once by the collective of these local minima.)

Observation 6. *Let* e *be an edge of* G*, and let* f, f' *be the two faces incident to* e *in* F^**. Then* $\lambda_t^*(f) \cap e = \lambda_t^*(f') \cap e$*.*

This observation says that wavefront edges of $\mathcal{W}_G$ of adjacent faces meet at the same point. This holds due to the bisector-condition.

Observation 7. *Let* v *be a vertex of* G *and let* $f_1, \dots, f_k$ *be the faces in cyclic order that are incident to* v *in* F^**. Then* v *has the same orthogonal distance to* $\lambda(f_i)$ *for all* $1 \leq i \leq k$*.*

Proof: For every incident edge e_i of v and the incident faces f_i, f_{i+1}, with $1 \leq i < k$ the bisector-condition implies that $\lambda(f_i)$ and $\lambda(f_{i+1})$ have the same orthogonal distance to v. ■

Lemma 8. *The initial wavefronts* $\mathcal{W}_G(\epsilon)$ *and* $\mathcal{W}_{\mathcal{S}(H)}(\epsilon)$ *are identical for* ϵ *so small that no vertex of* $\mathcal{S}(H)$ *or* G *has been hit by either wavefront at any time up to* ϵ*.*

Proof: The initial wavefront consists of copies of $\sigma(f)$ with $f \in F$ (and appropriate segments at degree-one vertices of H) that end at the edges of G resp. $\mathcal{S}(H)$ incident to H. In both cases, these edges are on the bisector of the corresponding edges of H. Hence the initial wavefronts are identical. ■

[1]We interpret the upper chain of f as the graph of a function, namely the orthogonal distance to $\lambda(f)$.

Lemma 9. *Assume that* $\mathcal{W}_G(t')$ *and* $\mathcal{W}_{\mathcal{S}(H)}(t')$ *are identical for* $0 < t' < t$*.*

- *If* $\mathcal{W}_G(t)$ *hits a vertex* v *of* G^**, then* v *coincides with a vertex of* $\mathcal{S}(H)$*.*
- *If* $\mathcal{W}_{\mathcal{S}(H)}(t)$ *hits a vertex* v *of* $\mathcal{S}(H)$*, then* v *coincides with a vertex of* G^**.*

Proof: The assumption $\mathcal{W}_G(t') = \mathcal{W}_{\mathcal{S}(H)}(t')$ for $0 < t' < t$ says that G and $\mathcal{S}(H)$ are identical at the locations that have been swept by the wavefronts for times $t' < t$. We will show the first claim, the same arguments apply to the second claim.

Let us denote by $f_1, \dots, f_k$, with $k \geq 3$, the faces of G^* incident to v. By Observation 7 the wavefronts sent out by $\sigma(f_1), \dots, \sigma(f_k)$ meet v simultaneously at time t. Let us fix i with $1 \leq i \leq k$ and consider a point sequence $(p^i)_n \in \lambda^*_{t \cdot (1 - \frac{1}{n})}(f_i) \cap f_i$ such that $v = \lim_{n \to \infty} (p^i)_n$. For every f_i there is a unique face g_i of $\mathcal{S}(H)$ with $\mathcal{W}_G(t') \cap g_i = \mathcal{W}_G(t') \cap f_i$, with $t' < t$. Note that $(p^i)_n \in g_i$ for all $n \in \mathbb{N}$ and as g_i is closed we have $v = \lim_{n \to \infty} (p^i)_n \in g_i$, for all $1 < i < k$. Hence $v \in \bigcap_{1 \leq i \leq k} g_i$, i.e., v is a vertex of $\mathcal{S}(H)$. ■

Definition 10. *Let* v *be a vertex of* G *(*$\mathcal{S}(H)$*, resp.). We denote an edge* e *of* v *as* ingoing to v *if it has been completely swept by the wavefront* $\mathcal{W}_G$ *(*$\mathcal{W}_{\mathcal{S}(H)}$*, resp.) when the wavefront reaches* v*. The other edges incident to* v *are called* outgoing from v*.*

Note that the outgoing edges are only touched at their endpoint v at the time when the wavefront reaches v.

Theorem 11. *Let* G *be a PSLG*$^\infty$*, for which all finite vertices have a degree of at least three. Then the mapping* λ *is a solution to Problem 1 if and only if* λ *fulfills the inside-, sweeping- and bisector-condition.*

Proof: In order to prove this theorem we show that $\mathcal{W}_G(t) = \mathcal{W}_{\mathcal{S}(H)}(t)$ for all times $t > 0$. The proof is by induction on the chronological order of vertices swept by the two wavefronts. These wavefronts trace the same edges until the first vertex v of G^* resp. $\mathcal{S}(H)$ is met. Lemma 9 allows us not to distinguish between these two cases. By the induction hypothesis, the ingoing edges of v are identical for G^* and $\mathcal{S}(H)$. The induction step claims that the outgoing edges of G^* and $\mathcal{S}(H)$ at v are identical in the local neighborhood of v, too. The proof of this claim concludes the entire proof.

We consider the local neighborhood of v, see Fig. 2. The shaded area depicts all points that were swept by the two wavefronts until time t. The first observation is that no outgoing edge e exists within the shaded area. Otherwise we could consider a locus on e that is swept twice by either wavefront, which is a contradiction as every locus of a each face of G^* resp. $\mathcal{S}(H)$ is swept only once.

From this observation it follows that if the neighborhood of v is entirely shaded then no outgoing edge exists and we are done. In other words, a connected component of the wavefront collapsed entirely at v.

In the general case, however, the local neighborhood of v is tessellated into shaded and non-shaded sectors. The key insight is that every non-shaded sector contains exactly one outgoing edge. This can be seen as follows.

Let us assume that a non-shaded sector contains no outgoing edge at all. However, the boundary of the unshaded sector belongs to two faces f_1 and f_2 of G^*. If $f_1 \neq f_2$ then, as the wavefront keeps on moving beyond time t. the two faces need to be separated in the neighborhood of v by some edge. If, on the other hand, $f_1 = f_2$ then v would constitute a local maximum in the lower chain of f_1 resp. f_2 as the two edges of f_1 that are incident to v have already been swept by the wavefront, see Fig. 2(a). This is a contradiction to the sweeping-condition. The same argument applies to $\mathcal{S}(H)$ and hence also $\mathcal{S}(H)$ needs to have an outgoing edge at v.

Let us now assume that two or more outgoing edges of G^* exist in a non-shaded sector, see Fig. 2(b). Hence, there is at least one face f' between two of the outgoing edges. Note that in a neighborhood of v the face f' has not been swept by the wavefront yet. In particular the edges of f' that are incident to v, have both not been swept by the wavefront yet. That is, v is a local minimum of the lower chain of f', which is a contradiction to the sweeping-condition. Again, the same argument applies to $\mathcal{S}(H)$, too.

Hence, there is exactly one outgoing edge e of G^* and e' of $\mathcal{S}(H)$, see Fig. 2(c) for the specific case of a split event. The edge e is on the common boundary of the faces f_1 and f_2 of G^* that together overlap the non-swept sector. Similarly, e' is on the common boundary of the faces g_1 and g_2 of $\mathcal{S}(H)$ that overlap the non-swept sector, too. However, as the wavefronts were identical until time $t' < t$ we know that g_1 is the straight-skeleton face of $\sigma(f_1)$ and g_2 is the straight-skeleton face of $\sigma(f_2)$. Hence, e and e' lie on the same bisector between $\lambda(f_1)$ and $\lambda(f_2)$. That is, in the local neighborhood of v the edges e and e' are identical. ■

As already mentioned, a straight-skeleton face fulfills a stronger version of the sweeping-condition: the lower chain is convex. By the previous theorem we were able to characterize the straight skeleton of a PSLG by the inside-, sweeping- and bisector-condition. Hence, it must be possible to show the convexity of the lower chain of a face of G^* using these three properties.

Lemma 12. *The lower chain of a face f of G^* is convex.*

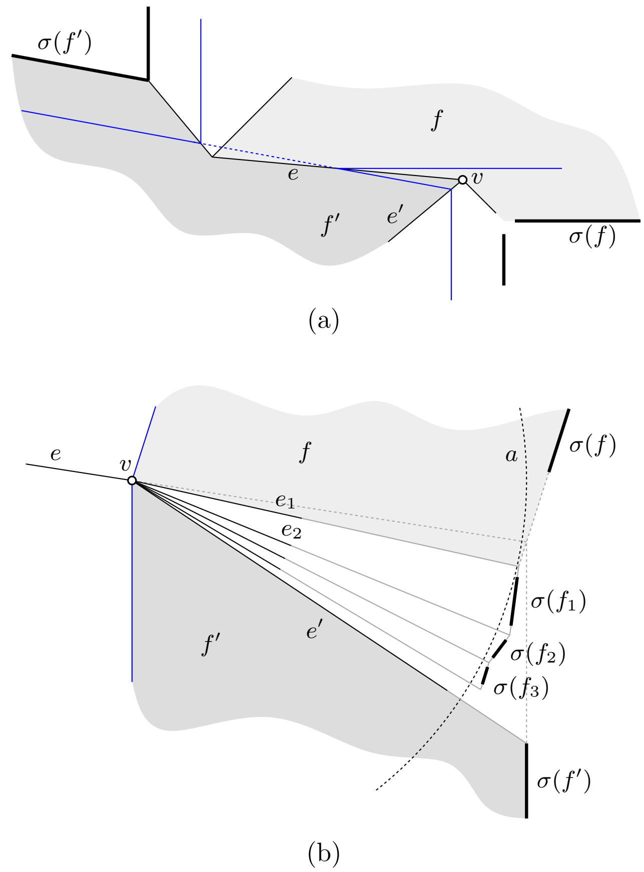

Fig. 3: The lower chain is convex.

Proof: Consider to the contrary that the lower chain of f has a vertex v that is reflex. We denote by $e \subset f$ the outgoing edge at v and by f' the opposite face of f at e. As e lies on the lower chain of f it also needs to lie on the lower chain of f' by the bisector-condition.

We first observe that e does not lie on the walk from v to $\sigma(f')$ along the lower chain of f', see Fig. 3(a). Otherwise, we would observe a local maximum on this lower chain as e is an outgoing edge at v, which is a contradiction to the sweeping condition.

We denote by e' (e_1, resp.) the second edge of f' (f, resp.) that is incident to v, respectively, see Fig. 3(b). The edge e' lies between e and $\sigma(f')$ on the lower chain and cannot be an outgoing edge of v by the same arguments given above. Hence, when the wavefront reaches v at time t the edges e_1 and e' have been completely swept. Furthermore, the lines $\lambda(f)$ and $\lambda(f')$ are tangential to the circle a that is centered at v and has radius t. We denote by f_1 the opposite face of f at e_1. The line $\lambda(f_1)$ also needs to be tangential to a as the wavefronts within f_1 reach v at the same time t. The face f_1 contains v and hence there needs to be an additional edge $e_2 \subseteq f_1$ that is incident to v. This edge e_2 does not lie on the bisector of $\lambda(f')$ and $\lambda(f_1)$ as the supporting line of e_2 does not intersect $\lambda(f') \cap \lambda(f_1)$. Hence, there is an additional face f_2 opposite to f_1 at e_2, and $\lambda(e_2)$ is tangential to a, too. We denote by e_3 the second edge of f_3 that is incident to v. By the same arguments above there is an additional face f_3, and so on. As there are only finitely many faces this yields a contradiction. ■

IV. Reconstructing straight skeletons

In this section, we solve Problem 1, i.e., given a PSLG$^\infty$ G, we determine whether there is another PSLG H such that the straight skeleton of H is G.

We note first that this problem is very easy if G has vertices of degree at most two: No vertices of degree zero or one can occur in a straight skeleton, and if there is a vertex of degree two in G, then it must have been caused by a vertex of degree one in H at the same point; and the slope of the unique edge of H at this vertex is determined. For this reason, we will throughout most of this section assume that G has minimum degree three; we briefly give the details of how to resolve vertices of degree two in Section IV-F.

A. Propagating lines and points

A key aspect of our recognition algorithm will be the propagation of lines $\lambda(f)$, for faces f of G, by successively reflecting them about edges of faces. In particular, if we know $\lambda(f)$ for one face f of a solution λ then we can obtain all other $\lambda(f')$ by recursively propagating $\lambda(f)$ along edges of G, i.e., along a spanning tree of the dual of G: We know that the bisector-condition must hold, and hence the line for face f determines uniquely the line in any other face f' that shares an edge with f. Similarly, we will propagate points just as lines along edges of G. We now introduce a formal notation for this propagation operation.

For any edge e of G and any line (point, resp.) l, define $\Phi_e(l)$ to be the line (point, resp.) obtained by reflecting l about the line supporting e. Put differently, $\Phi_e(l)$ is the unique line (point, resp) such that e is on the bisector of l and $\Phi_e(l)$. Yet another view is to "fold the paper" along the line through e, then $\Phi_e(l)$ is the line (point, resp.) where l is after folding.[2]

We will also propagate lines (points, resp.) along sequences of edges and hence extend Φ to directed walks in the natural way: $\Phi_\emptyset(l) := l$ and $\Phi_{e\circ W}(l) := \Phi_W(\Phi_e(l))$, with $e\circ W$ being a walk in the dual of G. We will specify a walk either explicitly by a sequence of edges of G or its dual, or implicitly by a sequence of faces of G such that two faces share an edge. (Note that if two faces share more than one edge then all those edges lie on the same line and hence the propagation step defined by Φ is the same for all such edges.) We also propagate sets of lines (points, resp.), and hence define $\Phi_W(S) := \bigcup_{l\in S}\Phi_W(l)$. Observe that for any propagation along a directed walk W the inverse operation is a propagation along the reverse walk W^{-1}.

B. The case of a star-graph

First consider the special case where the PSLG$^\infty$ G is a star, i.e., it has only one finite vertex v with, say, d rays $b_1, \dots, b_d$ to the vertex at infinity in cyclic order; see Fig. 4. Let β_i be the angle between rays b_i and b_{i+1} and, thus, $\beta_1 + \beta_2 \cdots + \beta_d = 2\pi$. Let f_i be the face incident to b_i and b_{i+1}. In the dual graph D, the faces $f_1, f_2, \dots, f_d, f_1$ form a cycle, which we denote by C. The goal is now to find a line $\lambda(f_i)$ for every face f_i that fulfills the inside-, bisector- and sweeping-condition. In particular, this implies that $\beta_i < \pi$ for all $1 \le i \le d$ and $\bigcup_{i=1}^d \sigma(f_i)$ forms a polygon whose vertices lie on the bisectors b_i. The essential condition, however, is the bisector-condition.

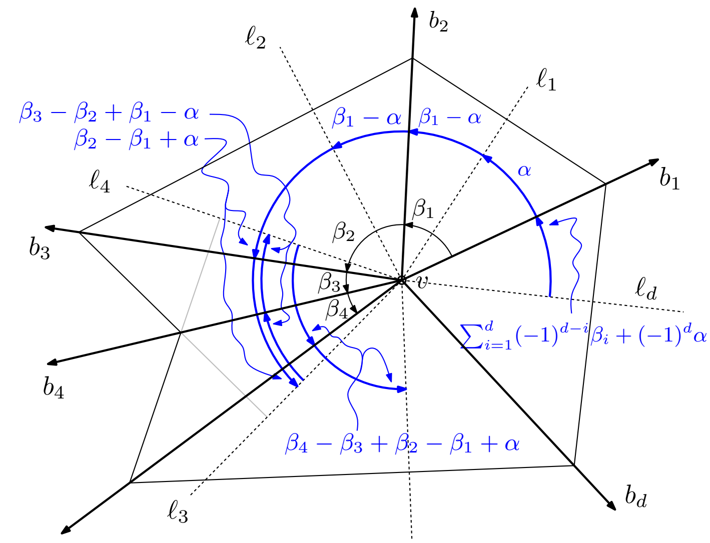

Fig. 4: The propagation of a line (shown thin) returns us to where we started if and only if $\alpha = \beta_d - \beta_{d-1} + \cdots + (-1)^{d-1}\beta_1 + (-1)^d\alpha$. Thick rays depict G, dashed lines indicate the lines $\ell(f, v)$.

In the following we will investigate the following question: Which lines l that intersect b_1 and b_2 and have a positive distance to v have the property that if we propagate them around v then we obtain the same line again? That is, which lines l fulfill the condition $\Phi_C(l) = l$? Note that if ℓ_1 is the line through v that is perpendicular to l then $\Phi_{b_2}(\ell_1)$ is again perpendicular to $\Phi_{b_2}(l)$. Hence, we can equivalently ask for all lines ℓ_1 through v intersecting the sector spanned by b_1 and b_2 such that $\Phi_C(\ell_1) = \ell_1$.

Let us denote by $\ell_{i+1} = \Phi_{b_1\circ b_2\circ\cdots\circ b_i}(\ell_1)$, with $1 \le i < d$, and denote by α the counter-clockwise angle between b_1 and ℓ_1. Then the angle between ℓ_1 and b_2 is $\beta_1 - \alpha$. By the bisector-condition b_2 is also the bisector between ℓ_1 and ℓ_2 and, hence, $\beta_1 - \alpha$ is also the angle between b_2 and ℓ_2. Hence, the angle between ℓ_2 and b_3 is $\beta_2 - \beta_1 + \alpha$ and so on. If we keep on adding up those angles, we see that the angle between ℓ_d and b_1 equals $\sum_{i=i}^d (-1)^{d-i}\beta_i + (-1)^d\alpha$. According to the bisector-condition at b_1 we therefore obtain

$$\alpha = \sum_{i=1}^d (-1)^{d-i}\beta_i + (-1)^d\alpha \tag{2}$$

and therefore

$$\frac{1}{2}\sum_{i=1}^d (-1)^{d-i}\beta_i = \begin{cases} 0 & \text{if } d \text{ is even,} \\ \alpha & \text{if } d \text{ is odd.} \end{cases} \tag{3}$$

Definition 13 (balance-condition)**.** *Let v be a finite vertex of G that has even degree d. We say that v satisfies the* balance-condition *if $\beta_d - \beta_{d-1} + \cdots + \beta_2 - \beta_1 = 0$.*

[2] Much of our algorithm to come could be expressed as "fold the paper along a spanning tree and see whether one needle can hit all layers". While this gives an intuitive idea, we prefer to express the algorithm via Φ instead to be able to verify correctness in detail.

Definition 14. *Let f be a face that has an incident vertex v of odd degree d, and let b_1, b_2 be the edges of f that are incident at v (in ccw order at v). Let $\beta_1, \dots, \beta_d$ be the angles at v in ccw order, starting with the angle at f. Define $\ell(f, v)$ to be the line that supports the ray from v at ccw angle $\frac{1}{2}(\beta_d - \beta_{d-1} + \beta_{d-2} - \cdots - \beta_2 + \beta_1)$ with b_1.*

Lemma 15. *If the degree of v is even then for any line ℓ_1 it holds that $\Phi_C(\ell_1) = \ell_1$ if and only if v fulfills the balance-condition. If the degree of v is odd then $\Phi_C(\ell_1) = \ell_1$ holds if and only if ℓ_1 is either $\ell(f_1, v)$ or perpendicular to $\ell(f_1, v)$.*

The previous lemma provides us with a simple algorithm to solve Problem 1 in case of a star graph G. In case that d is even we check whether the balance-condition holds. If it does not, then there is no solution and G is not a straight skeleton of any planar straight-line graph. But if the balance-condition holds then we choose any face f_1 incident to v and any line as $\lambda(f_1)$ that does intersect f_1 and has positive distance to v. In case that d is odd, we take any line as $\lambda(f_1)$ that is perpendicular to $\ell(f_1, v)$, intersects f_1 and has positive distance to v. If no such line exists then there is no solution.

In either case the other lines $\lambda(f_{i+1})$ are iteratively obtained by successively reflecting $\lambda(f_i)$ at b_i to obtain $\lambda(f_{i+1})$. Lemma 15 and Theorem 11 gives us the guarantee that the resulting graph H is a polygon whose straight skeleton is equal to G.

C. Arbitrary planar straight-line graphs

Now consider GMP-SS for an arbitrary input graph G. At any finite vertex v of G, we have the induced GMP-SS instance of v where we only use the edges incident to v (and extend them to be rays to a vertex at infinity). If GMP-SS has a solution, then so does the implied instance at v. Lemma 15 applied to the induced instances hence gives the following:

Lemma 16. *If GMP-SS(G) has a solution λ then at any finite vertex with even degree the balance-condition holds, and at any finite vertex v of odd degree and any face f incident to v, the solution-line $\lambda(f)$ is either identical or perpendicular to the line $\ell(f, v)$.*

We now summarize the constraints given by the vertices of odd degree of a face f by the set

$$\ell(f) := \{l \in \mathcal{L} : \ l \cap \operatorname{int} f \neq \emptyset\} \cap \bigcap_{\substack{v \text{ is vertex of } f \\ \deg(v) \text{ is odd}}} \{\ell(f, v)\} \cup \ell(f, v)^{\perp}, \qquad (4)$$

where $\ell(f, v)^{\perp}$ denotes all lines that are orthogonal to $\ell(f, v)$. By the previous corollary it follows for any solution λ that $\lambda(f) \in \ell(f)$ for all faces f.

The essential idea will now be to propagate these sets $\ell(f)$ of per-face solutions along edges of G to a single face, intersect all solutions, and propagate them back. For this purpose we fix an arbitrary spanning tree T of the dual D of G and root T at one vertex r^* of D, i.e., a face r of G. Furthermore, we denote by $i \rightsquigarrow^T j$ the unique path from vertex i to vertex j in T. Now define

$$\ell^r(f) := \Phi_{f \rightsquigarrow^T r}(\ell(f)) \qquad (5)$$

to be the propagation of $\ell(f)$ to the root-face r, and let

$$I := \bigcap_{f \in F} \ell^r(f). \qquad (6)$$

Note that for a line l and a face f it holds that

$$l \cap \operatorname{int} f \ \Leftrightarrow \Phi_{f \rightsquigarrow^T r}(l) \cap \Phi_{f \rightsquigarrow^T r}(\operatorname{int} f),$$

where f is interpreted as a point set, as $\Phi_{f \rightsquigarrow^T r}$ is a bijective map. Hence, we can express I as

$$I = \{l \in \mathcal{L} : \ l \cap \bigcap_{f \in F} \operatorname{int} f^r \neq \emptyset\} \cap \bigcap_{f \in F} \bigcap_{\substack{v \text{ is vertex of } f \\ \deg(v) \text{ is odd}}} \{\Phi_{f \rightsquigarrow^T r}(\ell(f, v))\} \cup (\Phi_{f \rightsquigarrow^T r}(\ell(f, v)))^{\perp}$$

A face f of G is said to be swept by l if the polygons that result from f after cutting it by l into parts all fulfill the sweeping-condition. (Those two polygons would constitute faces of G^*.)

Theorem 17. *Let G be a PSLG$^\infty$ where every finite vertex has degree three or more. Then GMP-SS(G) has a solution if and only if the balance-condition holds at all finite vertices of even degree, and there exists a line $l \in I$ such that for all faces $f \in F$*

- *$l \cap f^r$ is a single segment of finite length and*
- *l sweeps f^r.*

Moreover, the solutions of GMP-SS(G) are in one-to-one correspondence with such lines L.

Proof: Presume first that there is a solution λ to GMP-SS. By Lemma 16 the balance-condition holds at all finite vertices of even degrees. By Lemma 4, for any face f the line $\lambda(f)$ intersects f in one non-empty line segment and the intersection sweeps f. By the bisector-condition, propagating $\lambda(f)$ to r along $f \rightsquigarrow^T r$ gives $\lambda(r)$. Hence $\lambda(r) = \Phi_{f \rightsquigarrow^T r}(\lambda(f))$ intersects f^r in one line segment and $\lambda(r) \cap f^r$ sweeps f^r. Also if f has an incident vertex v of odd degree, then $\lambda(f)$ is identical or perpendicular to $\ell(f, v)$, and hence $\lambda(r)$ is identical or perpendicular to $\ell^r(f, v)$. This holds for all faces, so the line supporting $\lambda(r)$ satisfies all conditions. Any other solution to GMP-SS differs at least in one $\lambda(f)$ for a face f and, hence, has also a different $\lambda(r)$ and consequently gives a different line.

For the other direction, presume that l is such a line. Define $\lambda(f) := \Phi_{r \rightsquigarrow^T f}(l)$ for all faces f. By definition this satisfies the bisector condition for all edges whose dual is in T. By choice of l this also satisfies the inside-condition and the sweeping-condition for all faces.

It remains to show that any edge whose dual is a non-tree edge (f, f') bisects $\lambda(f)$ and $\lambda(f')$. Recall that $\lambda(f)$ and $\lambda(f')$ are obtained by propagating $\lambda(r)$ along $r \rightsquigarrow^T f$ and $r \rightsquigarrow^T f'$. If the bisector condition were violated at (f, f') then propagating $\lambda(f)$ along (f, f') would yield a line *different* from $\lambda(f')$. Put differently, propagating $\lambda(r)$ along the walk $(r \rightsquigarrow^T f) \circ (f, f')$ from r to f' yields a line *different* from $\lambda(f')$. We now show that this is impossible by proving a stronger claim:

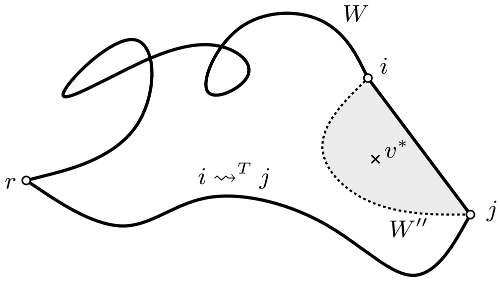

Fig. 5: By going around v the other way, we obtain a walk $W - (i, j) \cup W''$ that captures fewer faces.

Claim 18. *Let W be any walk in D from r to j. Then $\Phi_W(\lambda(r)) = \lambda(j)$.*

We prove the claim by induction on the number of faces of T captured by W, which is defined as follows. A closed walk is given by $W \cup (j \rightsquigarrow^T r)$. We say that a face v^* of T, i.e., a vertex v of G, is captured by[3] W if any path connecting v with the infinite vertex of G contains at least one edge of G whose dual belongs to W. We will also do an inner induction on the number of edges in W.

In the (outer) base case W captures no faces. Since any non-tree edge captures at least one face, W uses only tree-edges. Then either $W = r \rightsquigarrow^T j$, in case of which the claim holds by definition of $\lambda(j)$, or W contains a *U-turn* $k \to \ell \to k$ as a sub-path. Excising such a U-turn maintains the same propagation and shortens W and the claim holds by (inner) induction.

So, presume that W captures some faces. We first consider the case that W and $r \rightsquigarrow^T j$ share some intermediate vertex, say $W = r \rightsquigarrow^{W'} x \rightsquigarrow^{W''} j$ with W' and W'' non-empty and x a vertex in $r \rightsquigarrow^T j$. Apply the claim first to W' to show $\Phi_{W'}(\lambda(r)) = \lambda(x)$, and then to $(r \rightsquigarrow^T x) \cup W''$ to show $\Phi_{W''}(\lambda(x)) = \lambda(j)$. This can be done since W' and W'' capture no more faces than W and are shorter. Combining the two results yields the claim in this case.

We consider now the case that W and $r \rightsquigarrow^T j$ are disjoint except at the ends. Let (i, j) be the last edge of W, i.e., $W = r \rightsquigarrow^{W'} i \to j$. Let v^* be a face of D that is adjacent to (i, j) and captured by W. (This exists since W does not visit j earlier.) See also Fig. 5. Face v^* of D corresponds to a vertex v in G, which is finite since v^* is captured by W. Let W'' be such that $j \to i \rightsquigarrow^{W''} j$ is the cycle around v^*. Observe that propagating $\lambda(j)$ along $j \to i \rightsquigarrow^{W''} j$ returns us to $\lambda(j)$, for we propagate $\lambda(j)$ around vertex v, and either the balance-condition holds at v or $\lambda(j) \in \{\ell(j, v)\} \cup \ell(j, v)^\perp$ by choice of $\lambda(r)$. Inverting the propagation therefore gives $\Phi_{(W'')^{-1} \circ (i,j)}(\lambda(j)) = \lambda(j)$.

Now apply induction to $W' \circ W''$, i.e., we avoid edge (i, j) and instead go "the other way" around face v^*. Since face v^* was captured by W, it is not captured by $W' \circ W''$, and hence we can apply induction and know that $\Phi_{W' \circ W''}(\lambda(r)) = \lambda(j)$.

[3]"Captured by $W \cup (j \rightsquigarrow^T r)$" would be more precise, but is cumbersome to write and not necessary since W determines $j \rightsquigarrow^T r$ via its endpoints and the fixed tree T.

Combining this with the propagation around v yields

$$\begin{aligned} \Phi_W(\lambda(r)) &= \Phi_{W' \circ W'' \circ (W'')^{-1} \circ (i,j)}(\lambda(r)) \\ &= \Phi_{W' \circ W''}(\lambda(j)) \\ &= \lambda(j) \end{aligned}$$

as desired. ■

D. Algorithm and run-time

In this section, we will turn the previous proof into an algorithm that solves GMP-SS. We can compute the propagation function and compute the propagated faces f^r and lines $\ell^r(f, v)$ for all faces f and incident vertices v in total linear time. It now remains to find one line l such that l intersects all f^r in one non-empty line segment s that sweeps f^r, and that also is perpendicular to any line $\ell^r(f, v)$ that may exist.

We distinguish two cases. In the first case, there exists no line $\ell(f, v)$, i.e., all vertices of G have even degree. Observe that in this case all faces of G are convex, as any angle greater than π makes the balance-condition impossible to satisfy. But if f^r is convex then *any* line that intersects f^r intersects it in a single line segment, and this line segment trivially sweeps f^r. Therefore, to satisfy the conditions we only have to find a line l that intersects all convex polygons f^r. (Recall that all convex regions f^r are bounded since we restrict our search for a solution to a finite subset of the plane.) Such a line can be determined in $O(n \log n)$ time [12], [13], and with some modifications the algorithm by [12] can also find all such lines in $O(n \log n)$ time.

In the second case, there exists at least one line $\ell(f, v)$ to which our line l must be identical or perpendicular. If these lines are not all parallel or perpendicular, then there is no solution. If all these lines are parallel or perpendicular, then there are only two possible slopes of l. Say (after possible rotation) that l must be horizontal or vertical. We can test whether a suitable horizontal l by distinguishing the following cases, which can be distinguished in linear time:

- f^r is x-monotone. We can find the lowest local minimum on the upper chain and the highest local maximum on the lower chain. Line l is suitable for f^r if and only if it intersects f^r somewhere between these two vertices.
- f^r is not x-monotone. We need to find a horizontal line l that partitions f^r into two x-monotone polygons even tough f^r is not x-monotone, see Fig. 6. Let L denote a vertical sweep-line. If f^r is swept by a horizontal line l then the sequence of the number of segments of $L \cap f^r$ when L moves from left to right is $1, 2, 1, 2, 1$. (Dark shaded area depicts the location where $L \cap f^r$ consists of two segments.) Let u be the vertex where the two segments of $L \cap f^r$ merge to one and let v be the vertex where the one segment of $L \cap f^r$ splits into two again, see Fig. 6(a). We can find the vertices u and v in linear time. The only suitable horizontal line l is the one on which u and v lie. If u and v are have different y-coordinates then no suitable line l exists. If u (resp. v) does not exist then the only suitable candidate for l is the horizontal

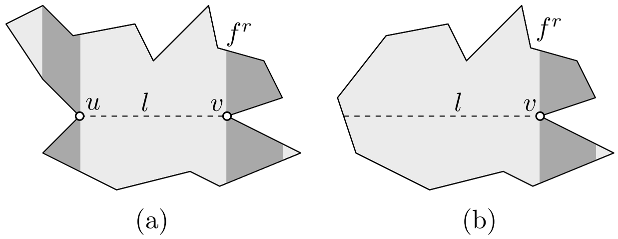

Fig. 6: The line l partitions f^r into two x-monotone faces.

line that supports v (resp. u), see Fig. 6(b). If both u and v do not exist then f^r is x-monotone.

Hence each face f^r defines an interval (possibly containing only a single point) for the abscissa of l, and intersecting these intervals (which can be done in $O(n)$ time) gives the range of possible horizontal lines l. Similarly we can determine the range of vertical lines l.

E. Avoiding trigonometric calculations

The obvious way in order to implement our algorithm involves the usage of trigonometric functions and their inverse. We sketch in this section how all such trigonometric computations can be avoided, presuming we can handle real numbers and arithmetic operations on them (including square-root) in constant time.

We presume that our input is specified by giving coordinates for every finite vertex, as well as one additional point for each ray and two additional points for each line.

We illustrate the approach only for how to check the balance condition at one vertex v of even degree d; all other operations can be handled similarly. Let $b_1, \ldots, b_d$ be the edges incident to v in ccw order. For ease of description, let us assume that v is at the origin and b_1 lies on the positive x-axis (if this is not the case then we break up the angle that contains the positive x-axis into two and apply a similar argument.) For each b_i, we know one other point p_i that is on b_i and not at the origin (either the other endpoint, if b_i is a line segment, or one other point if b_i is a ray.)

Now observe that the coordinates of $p_i/\|p_i\|$ are $\cos(\beta_1 + \cdots + \beta_{i-1})$ and $\sin(\beta_1 + \cdots + \beta_{i-1})$. Hence for each i from the coordinates of p_i we can compute $\cos(\beta_1 + \cdots + \beta_{i-1})$ and $\sin(\beta_1 + \cdots + \beta_{i-1})$ in constant time (this involves square roots). After this, we can compute

$$\begin{aligned}\cos(\beta_i) &= \cos\left((\beta_1 + \cdots + \beta_i) - (\beta_1 + \cdots + \beta_{i-1})\right) \\ &= \cos\left(\beta_1 + \cdots + \beta_i\right)\sin\left(\beta_1 + \cdots + \beta_{i-1}\right) \\ &\quad + \sin\left(\beta_1 + \cdots + \beta_i\right)\cos\left(\beta_1 + \cdots + \beta_{i-1}\right)\end{aligned}$$

in constant time per index i since all values on the right-hand side are known. Similarly we compute $\sin(\beta_i)$ for all i; this takes $O(d)$ time in total.

Define $B_i = \beta_1 - \beta_2 + \beta_3 - \cdots + (-1)^{i-1}\beta_i$. Observe that $B_i = B_{i-1} + (-1)^{i-1}\beta_i$, and using a similar trick as above, we can hence compute $\cos(B_i)$ and $\sin(B_i)$ for all i in $O(d)$ time. But the balance-condition at v holds if and only if $B_d = 0$, if and only if $\sin B_d = 0$ and $\cos B_d = 1$, so after computing $\sin B_d$ and $\cos B_d$ we can check the balance-condition in constant time.

F. Vertices of degree two

It remains to consider the special case of a vertex v of degree two in G. This can happen if and only if the input PSLG H has a vertex of degree one at v. Moreover, the straight skeleton must have one angle of $\pi/2$ at v, and the unique incident edge of v in H must bisect the $3\pi/2$ angle at v.

Hence, if G has a vertex v of degree two then we first test whether the angles at v are $\pi/2$ and $3\pi/2$; no solution exists otherwise. If the angles are correct then in any solution the face f at the $3\pi/2$ angle must use the line $\lambda(f)$ that bisects the angle. Note that the bisector-condition must hold even if there are vertices of degree two. Hence we can propagate $\lambda(f)$ to all faces (along any spanning tree of the dual graph) to obtain the only possible solution-candidate $\lambda(.)$. Now verify whether with this $\lambda(.)$ the bisector-condition holds for all non-tree edges, as well as the inside-condition and sweeping-condition for all faces that are not incident to a $\pi/2$ angle at a vertex of degree two. It is not hard to show (details are omitted) that this holds if and only if $\lambda(.)$ is a solution to GMP-SS(G).

Putting it all together, we therefore have our main theorem:

Theorem 19. *GMP-SS(G) can be solved and the set of feasible solutions can be found in $O(n \log n)$ time for an input PSLG$^\infty$ G with n edges in the Real RAM model of computation.*

V. Recognizing Voronoi diagrams

We now employ the same strategy used for solving GMP-SS to recognize and reconstruct Voronoi diagrams of points, without restriction on the degrees of G, thus adding the missing part to the solution proposed by Ash and Bolker [6].

Problem 4 (GMP-VD)**.** *Given a PSLG$^\infty$ G, can we find a (finite) set S of points such that $\mathcal{VD}(S) = G$?*

A solution to GMP-VD consists of a mapping $\rho : F \to \mathbb{R}^2$ of faces to points such that the Voronoi diagram of these points equals G. The following lemma is proved easily. (See, e.g., [6].)

Lemma 20. *Let $\rho : F \to \mathbb{R}^2$ be a mapping from faces of a PSLG$^\infty$ G to points in the plane. Then ρ is a solution to GMP-VD if and only if the following two conditions hold:*

1) *Inside-condition: $\rho(f)$ is strictly inside f for any face $f \in F$.*
2) *Bisector-condition: For any edge e of G with incident faces f and f', the line through e is a bisector of $\rho(f)$ and $\rho(f')$.*

Suppose that G is a star graph, with one finite vertex v. Let $f_1, f_2, \ldots, f_d$ be the faces around v In the dual graph of G these faces form a cycle, which we denote by C. A mapping ρ from faces of G to points satisfies the bisector-condition if and only if propagating the point $\rho(f_1)$ around v brings us back to where we started. In terms of our formal propagation notation, we need that $\Phi_C(\rho(f_1)) = \rho(f_1)$.

Lemma 21. *If the degree of v is even then for any point p in f_1 it holds that $\Phi_C(p) = p$ if and only if v fulfills the balance-condition (Def. 13). If the degree of v is odd then $\Phi_C(p) = p$ holds if and only if p is on $\ell(f_1, v)$ (Def. 14).*

As for straight skeletons, the induced instances give the following lemma.

Lemma 22. *If GMP-VD(G) has a solution ρ then at any finite vertex of G with even degree the balance-condition holds, and at any finite vertex v of odd degree and any face f incident to v, the solution-point $\rho(f)$ lies on the line $\ell(f, v)$.*

Now define for any face f the set

$$S(f) := (\operatorname{int} f) \cap \bigcap_{\substack{v \text{ is vertex of } f \\ \deg(v) \text{ is odd}}} \ell(f, v).$$

If a face has vertices of even degree only then $S(f) := \operatorname{int} f$. By the inside-condition and Lemma 22, if GMP-VD has a solution ρ then for any face f the point $\rho(f)$ must be in $S(f)$.

Notice in particular that if a face has two vertices v_1, v_2 of odd degree for which the lines $\ell(f, v_1)$ and $\ell(f, v_2)$ are not parallel then $S(f)$ consists of only one point. Once one point of a face is fixed, propagating it to all other faces gives the only possible candidate for a Voronoi diagram input. Hence, as pointed out by Ash and Bolker [6], GMP-VD is easily solvable if all vertices have odd degree (as would be the case in Voronoi diagrams of points in general position.)

To solve GMP-VD for an arbitrary graph G, we again fix a spanning tree T of the dual graph D of G and root it arbitrarily at one vertex r of D, i.e., face r of G. We define

$$S_r(f) := \Phi_{f \rightsquigarrow_T r}(S(f))$$

to be the propagation of $S(f)$ to the root-face r, and let

$$I := \bigcap_{f \in F} S_r(f) = \bigcap_{f \in F} \Phi_{f \rightsquigarrow_T r}(S(f)).$$

Again, one can show that the result does not depend on the particular spanning tree T of D chosen for the propagation, and we get the following theorem:

Theorem 23. *GMP-VD(G) has a solution if and only if the balance-condition holds at all finite vertices of even degrees of G and I is non-empty. Moreover, the solutions of GMP-VD(G) are in one-to-one correspondence with the points in I.*

We can compute the intersection I of the propagated half planes of G in $O(n \log n)$ time, or test whether it is non-empty in $O(n)$ time. (This is a standard computational geometry problem, see for example [14].) If I contains a point p then we compute $\rho(f)$ by applying $\Phi_{r \rightsquigarrow_T f}$ to p, and output this as solution to GMP-VD(G). Similar considerations as in Sec. IV-E allows us to implement this algorithm without trigonometric operations. We summarize our result in the following theorem:

Theorem 24. *GMP-VD can be solved in $O(n)$ time, and the set of feasible solutions can be described by a convex set of points that can be computed in $O(n \log n)$ time, under the Real RAM model of computation.*

VI. Conclusion

In this paper, we considered the problem of reconstructing the input if we are given a straight skeleton. We showed that we can test efficiently whether a given structure is indeed a straight skeleton, and that we can characterize all possible inputs. Our algorithms operate in the Real RAM computer model and take $O(n \log n)$ time. A similar approach can also be applied if the input allegedly is a Voronoi diagram of points, but in this case the mere existence of a solution can be tested in $O(n)$ time.

For the straight skeleton, we only considered the case of a straight skeleton of a PSLG. We note here that our algorithm can be adapted to handle straight skeletons of polygons as well. Specifically, we can answer the following question: Given a PSLG$^\infty$ G and a set of vertex V' of G, is there a polygon P whose straight skeleton has vertices V' and coincides with $G \cap P$? In essence this is done by checking the conditions only for faces incident to vertices at V'; we omit the details.

As for open problems, we would be interested in answering the same questions under further restrictions on the input. For example, how easy is it to test whether a given PSLG$^\infty$ is the straight skeleton of a monotone polygon? In particular, which properties characterize the straight skeleton of a monotone polygon?

References

[1] S. Huber and M. Held, "A Fast Straight-Skeleton Algorithm Based on Generalized Motorcycle Graphs," *Internat. J. Comput. Geom. Appl.*, vol. 22, no. 5, pp. 471–498, Oct. 2012.

[2] M. Dillencourt, "Realizability of Delaunay Triangulations." *Inform. Process. Lett.*, vol. 33, no. 6, pp. 283–287, 1990.

[3] M. B. Dillencourt and W. D. Smith, "Graph-Theoretical Conditions for Inscribability and Delaunay Realizability," *Discrete Math.*, vol. 161, no. 1-3, pp. 63–77, 1996.

[4] G. Liotta and H. Meijer, "Voronoi Drawings of Trees," *Comput. Geom. Theory and Appl.*, vol. 24, no. 3, pp. 147–178, 2003.

[5] O. Aichholzer, H. Cheng, S. Devadoss, T. Hackl, S. Huber, B. Li, and A. Risteski, "What Makes a Tree a Straight Skeleton?" in *Proc. 24th Canad. Conf. Comput. Geom.(CCCG'12)*, Charlottetown, P.E.I., Canada, Aug. 2012, pp. 267–272.

[6] P. Ash and E. Bolker, "Recognizing Dirichlet Tesselations," *Geometriae Dedicata*, vol. 19, pp. 175–206, 1985.

[7] D. Hartvigsen, "Recognizing Voronoi Diagrams with Linear Programming," *ORSA J. Computing*, vol. 4, no. 4, pp. 369–374, 1992.

[8] F. Aurenhammer, "Recognizing Polytopical Cell Complexes and Constructing Projection Polyhedra," *J. Symbolic Comput.*, vol. 3, no. 3, pp. 249–255, 1987.

[9] T. Biedl, M. Held, and S. Huber, "Reconstructing Polygons from Embedded Straight Skeletons," in *Proc. 29th Europ. Workshop Comput. Geom.*, Braunschweig, Germany, Mar. 2013, pp. 95–98.

[10] O. Aichholzer, D. Alberts, F. Aurenhammer, and B. Gärtner, "Straight Skeletons of Simple Polygons," in *Proc. 4th Internat. Symp. of LIESMARS*, Wuhan, P.R. China, 1995, pp. 114–124.

[11] S. Huber, *Computing Straight Skeletons and Motorcycle Graphs: Theory and Practice*. Shaker Verlag, Apr. 2012, iSBN 978-3-8440-0938-5.

[12] J.-M. Robert and G. Toussaint, "Computational Geometry and Facility Location," in *Proc. Int. Conf. Operations Research and Management Science*, Manila, The Philippines, Digital Press 1990, pp. B.1–B.19. [Online]. Available: http://cgm.cs.mcgill.ca/~orm/thstrip.html

[13] B. Bhattacharya, J. Czyzowicz, P. Egyed, G. Toussaint, I. Stojmenovic, and J. Urrutia, "Computing Shortest Transversals of Sets," *Internat. J. Comput. Geom. Appl.*, vol. 2, no. 4, pp. 417–442, 1992.

[14] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry. Algorithms and Applications*, 3rd ed. Springer-Verlag, 2008, iSBN 978-3-540-77973-5.