Theoretical and Practical Results on Straight Skeletons of Planar Straight-Line Graphs*

Stefan Huber
FB Computerwissenschaften
Universität Salzburg
5020 Salzburg, Austria
shuber@cosy.sbg.ac.at

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

ABSTRACT

We study straight skeletons and make both theoretical and practical contributions which support new approaches to the computation of straight skeletons of arbitrary planar straight-line graphs (PSLGs). We start with an adequate extension of the concept of motorcycle graphs to PSLGs, with motorcycles starting at the reflex vertices of a PSLG, which allows us to generalize well-known results on the relation between the straight skeleton and the motorcycle graph to arbitrary PSLGs: the edges of the motorcycle graph cover a specific subset of the edges of the straight skeleton, and they form the basis of 3D slabs such that the projection of the lower envelope of those slabs to the plane forms the straight skeleton. As an immediate application we sketch how to use a graphics hardware for computing (approximate) straight skeletons of PSLGs. Further, we present and analyze a novel wavefront-type algorithm which bridges the current gap between the theory and practice of straight-skeleton computations. Our algorithm handles arbitrary PSLGs, is easy to implement, and is fast enough to handle complex data: it can be expected to run in $O(n \log n)$ time in practice for an n-vertex PSLG; its worst-case complexity is $O(n^2 \log n)$. Extensive experimental results confirm an average runtime of $20n \log n\ \mu s$ on a standard PC for virtually all of our 13 500 datasets of different characteristics. As also confirmed by our experiments, this constitutes an average gain in performance by a multiplicative factor of n, or at least one to two orders of magnitude, relative to the speed of the implementation provided by CGAL for closed polygons.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

General Terms

Algorithms, Theory

Keywords

Straight skeleton, motorcycle graph, wavefront propagation, implementation, experiments

*Work supported by Austrian FWF Grant L367-N15.

SCG'11, June 13–15, 2011, Paris, France.

1. INTRODUCTION

1.1 Motivation and prior work

This paper deals with new theoretical and practical results on straight skeletons of planar straight-line graphs (PSLGs). Aichholzer et al. [1] introduced the straight skeleton of a simple polygon as a skeleton structure similar to Voronoi diagrams. Roughly speaking, the straight skeleton of a simple polygon is defined by a wavefront propagation process where the edges move inwards in a self-parallel manner. The traces of the moving vertices of this process form the straight skeleton. The generalization of this concept to PSLGs was carried out by Aichholzer and Aurenhammer [2, 3].

Four different types of algorithms for computing straight skeletons are known. Aichholzer et al. [1] presented an algorithm for polygons as input which runs in $O(n^2 \log n)$ time for an n-vertex polygon. Basically, it simulates the propagating wavefront. Cheng and Vigneron [6] introduced an algorithm with an expected runtime of $O(n\sqrt{n} \log^2 n)$ for polygons with holes, where so-called vertex events[1] may not occur. Unfortunately, this algorithm is too complex to be implemented and, indeed, no implementation is known. Aichholzer and Aurenhammer [3] presented a wavefront-type algorithm based on triangulations which accepts PSLGs as input. Currently, $O(n^3 \log n)$ is the best bound known for its worst-case complexity. The authors report an $O(n \log n)$ runtime for sample runs of their code on real-world data but no extensive experimental results are known. Eppstein and Erickson [7] introduced an $O(n^{17/11+\epsilon})$ algorithm for simple polygons which could also be applied to PSLGs. However, also their algorithm is too complex to be implemented such that the theoretical runtime complexity is achieved in practice.

The only remaining contender is the implementation by Cacciola [5], as part of CGAL. His implementation takes a simple polygon with holes as input; it is roughly based on a wavefront-type algorithm by Felkel and Obdržálek [8]. The

[1]Vertex events are events where two or more reflex vertices meet during the wavefront propagation.

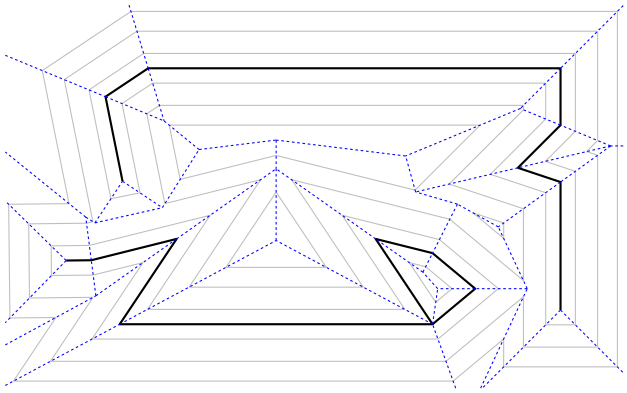

Figure 1: The straight skeleton (dashed) is defined by propagating wavefronts (grey) of the input (bold).

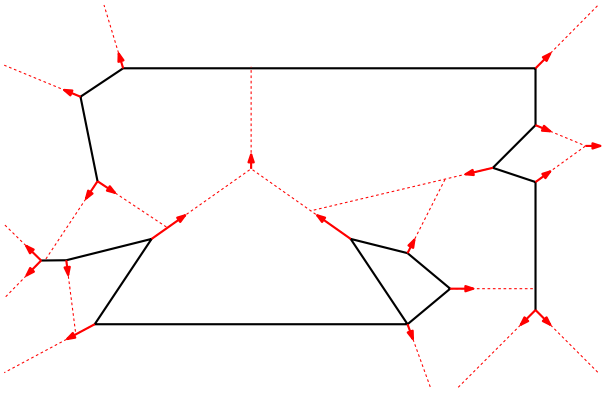

Figure 2: The motorcycle graph (dashed) is defined by the motorcycles and walls. Every arrow depicts the speed vector of a motorcycle. Walls are shown bold.

worst-case complexity of the original algorithm is $O(n^2)$; our experiments show that the average runtime of the CGAL code is at least $\Omega(n^2)$.

1.2 Our contribution

First, we present new theoretical results on straight skeletons of PSLGs in Section 2. In particular we present extensions of theorems of Cheng and Vigneron [6] and Eppstein and Erickson [7]. For a properly generalized motorcycle graph defined by the reflex vertices of the PSLGs, we obtain that the edges of the motorcycle graph cover a specific subset of the edges of the straight skeleton, and that they form the basis of 3D slabs such that the projection of the lower envelope of those slabs to the plane forms the straight skeleton. Note that our results hold also if vertex events occur. We develop two practical implications: (i) first we briefly discuss a straight-skeleton algorithm based on rendering by means of graphics hardware, and (ii) in Section 3 we present a new wavefront-type straight-skeleton algorithm.

The results described in Section 2 allow us to employ the motorcycle graph in order to improve the classical wavefront-type algorithm by Aichholzer et al. [3]. Our algorithm is easy to implement and fast enough to be applicable to complex real-world data: it exhibits an $O(n \log n)$ runtime for thousands of real-world and synthetic datasets, and has a worst-case complexity of $O(n^2 \log n)$. Finally, we present extensive experimental results in Section 4 where we compare the runtime of our implementation against the implementation of CGAL. As it turns out, our implementation achieves on average a performance boost of one to two orders of magnitude, or of a multiplicative factor of n, for moderate-sized datasets of up to 10 000 vertices. Moreover, in contrast to the CGAL implementation, our code is able to handle datasets with a million (or more) vertices on a standard PC. Thus, the implementation of our algorithm is the first code which is capable of handling complex real-world datasets.

1.3 Preliminaries

Consider a planar straight-line graph G with n vertices, none of them being isolated. Vertices of degree one are called terminals. According to [3], the definition of the straight skeleton $\mathcal{S}(G)$ of G is based on a wavefront-propagation process: Every edge e of G sends out two wavefronts, which are parallel to e and have unit speed. At terminals of G an additional wavefront orthogonal to the single incident edge is emitted. The wavefront $\mathcal{W}(G,t)$ of G at some time t can be interpreted as a 2-regular kinetic straight-line graph. Except for the vertices originating from the terminals of G, all vertices of $\mathcal{W}(G,t)$ move along bisectors of straight-line edges of G, see Fig. 1. During the propagation of $\mathcal{W}(G,t)$ topological changes occur: an edge may collapse ("edge event") or an edge may be split by a vertex ("split event"). The straight-line segments traced out by the vertices of $\mathcal{W}(G,t)$ form the so-called "arcs" (according to [3]) of $\mathcal{S}(G)$.

Aichholzer and Aurenhammer [3] gave a powerful interpretation of $\mathcal{S}(G)$. They considered a fixed-slope terrain in $\mathbb{R}^3$ by treating time as a third spatial dimension: they embed G and $\mathcal{S}(G)$ in the ground plane $\mathbb{R}^2 \times \{0\}$. Now assume that the propagating wavefronts $\mathcal{W}(G,t)$ of G are moving upwards at unit speed. Then the wavefronts form a fixed-slope terrain $\mathcal{T}(G) \subset \mathbb{R}^3$ of the form $\bigcup_{t \geq 0} \mathcal{W}(G,t) \times \{t\}$. The wavefront at some time t can be interpreted as the isoline of $\mathcal{T}(G)$ at height t. The straight skeleton $\mathcal{S}(G)$ is given by the projection of the valleys and ridges of $\mathcal{T}(G)$ onto the ground plane.

We call an arc e of $\mathcal{S}(G)$ reflex (convex, resp.) if the corresponding edge $\hat{e}$ in $\mathcal{T}(G)$ is a valley (ridge, resp.). Further, we call a vertex v of $\mathcal{W}(G,t)$ reflex (convex, resp.) if the angle between the two incident edges on the side where v propagates to is $\geq 180^o$ ($< 180^o$, resp.). Hence, reflex arcs of $\mathcal{S}(G)$ are traced out by reflex vertices of $\mathcal{W}(G,t)$, and vice versa.

Consider a set of points in the plane, called "motorcycles", that drive along straight-line rays according to given speed vectors and a specific start time. Further consider a set of straight-line segments, called "walls". Every motorcycle leaves a trace behind it and stops driving—it "crashes"—when reaching the trace of another motorcycle or a wall. The arrangement of these traces is called motorcycle graph, cf. [10]. In Fig. 2 we plot the motorcycle graph for a specific set of motorcycles motivated in the next section. The motorcycle graph problem was introduced by Eppstein and Erickson [7]. In their original formulation all motorcycles had to start at the same time and no walls were considered.

2. MOTORCYCLE GRAPH INDUCED BY A PSLG

We start by defining the motorcycle graph $M(G)$ which is induced by a planar straight-line graph G. In order to do so we have to specify the set of walls and for each motorcycle its start point, start time and speed vector.

First of all, the walls are given by the straight-line edges of G. Every motorcycle is defined by two propagating edges e and e' of the wavefront. We denote by $e(t)$ the straight-line segment of a wavefront edge e at time t and by $\overline{e(t)}$ its supporting line. Similarly, $v(t)$ denotes the wavefront vertex v at time t. For every reflex vertex v of $\mathcal{W}(G,0)$ we consider a motorcycle m which starts at time 0 and is defined by the two incident wavefront edges e, e' of v as follows: The position of m at time t, denoted by $m(t)$, is given by the intersection point $\overline{e(t)} \cap \overline{e'(t)}$. This implies that the start point of m is exactly $v(0)$. Note that at every terminal of G two motorcycles are launched, see Fig. 3 (a–b). We call the edge left of the trace of m the left arm of the motorcycle m, and the edge right of it the right arm. By the angle between the two arms we mean the angle on the side where the motorcycle move to.

If two or more motorcycles crash simultaneously at a point p then we first consider a local disc D around p. This disc is tessellated into slices by the motorcycle traces established up to the current simulation time. If one of those slices is non-convex, see Figure 3 (c-d), then we start a new motorcycle which is defined by the following procedure. Denote by $m_1, m_2, \ldots, m_k$ the motorcycles that crashed at p such that (i) their corresponding traces appear counter-clockwise around p and (ii) the traces of m_1 and m_k bound the non-convex slice. We distinguish the following two cases:

1. The left arm of m_1 and the right arm of m_k span a reflex angle: Then we start at p a new motorcycle m' which is defined by the left arm of m_1 and the right arm of m_k. Fig. 3 (c) illustrates this situation.

2. The left arm of m_1 and the right arm of m_k span a convex angle: Then we shoot from p a new motorcycle m' which continues the movement of m_k. That is, m' inherits its left and right arm from m_k, see Fig. 3 (d). (Letting a motorcycle continue in this way is a technical twist in order to avoid that the arms of m' span a convex angle on the propagation side of m'; it is used in the proofs of Lem. 1 and Thm. 2.)

In both cases we call $m_1, \ldots, m_k$ the ancestors of m'. In particular, m_1 is its left-most ancestor and m_k is its right-most ancestor. The right-most ancestor chain of m' consists of m', the right-most ancestor m_k of m', the right-most ancestor of m_k, and so on. Likewise we define the left-most ancestor chain of m'.

We denote by $\mathcal{M}(G)$ the motorcycle graph that results from the setup given above. Fig. 2 depicts $\mathcal{M}(G)$ for the sample PSLG G of Fig. 1. The basic idea behind the definition of the motorcycles is that every reflex arc of $\mathcal{S}(G)$ should be covered by a motorcycle trace and that the speed of the reflex wavefront vertex should correspond to the speed of the corresponding motorcycle. A simultaneous crash of several motorcycles corresponds to the analogous straight skeleton case of so-called "vertex events" ("multi split events", see Sec. 3), where multiple reflex arcs end in a common endpoint.

The following two assertions, Lemma 1 and Theorem 2, constitute extensions of results by Cheng and Vigneron [6] and Eppstein and Erickson [7]; they are essential for our straight skeleton algorithm. In the sequel we often use the notation "tilted motorcycle $\hat{m}$" to put m into the terrain notation of the straight skeleton by considering the z-axis as the time axis while m moves, cf. [7]. (The slope of $\hat{m}$ corresponds to the reciprocal value of the speed of m.)

LEMMA 1. *Consider a point p of $\mathcal{M}(G)$ which does not coincide with G. Then a local disc around p is tessellated into convex slices by $\mathcal{M}(G)$.*

Proof omitted due to lack of space.

THEOREM 2. *The reflex arcs of $\mathcal{S}(G)$ are covered by $\mathcal{M}(G)$.*

PROOF. The proof consists of two steps. In the first step we state and prove an essential claim which is applied later on in a proof by contradiction of the original theorem.

(1) Let m be a motorcycle and $p \in \mathbb{R}^2$ a point on the trace of m. If all valleys of $\mathcal{T}(G)$ are covered by (tilted) motorcycle traces up to the height of $\hat{m}$ at p, then the height of $\hat{m}$ is greater or equal to the height of $\mathcal{T}(G)$ at p. Equality is attained if and only if the valley of $\mathcal{T}(G)$, which corresponds to m, exists until p.

We denote by e the right arm of m and denote by $m_1, \ldots, m_k$ the right-most ancestor chain of m such that $m_1(0)$ is incident to $e(0)$ and m_k equals m, see Fig. 4 for $k = 2$. Further, we denote by p_i the endpoint of the trace of m_i. We arrive at the following observations:

- The motorcycles $m_1, \ldots, m_k$ share the same right arm e. As a consequence, the tilted traces $\hat{m}_1, \ldots, \hat{m}_k$ lie on a plane, namely the supporting plane of the terrain face that corresponds to the wavefront edge e.

- The angles between $e(0)$ and the trace of m_1, between m_1 and m_2, and so on are convex by Lem. 1. However, the angle between $e(0)$ and the trace of m_i, with $1 \leq i \leq k$, is at least 90^o by definition of the motorcycles. (Assume that $e(0)$ and the trace of m_i are extended until they meet.)

Let us denote by $\mathcal{T}$ the polygonal chain that is defined by the intersection of $\mathcal{T}(G)$ with a vertical curtain whose lower boundary (in the ground plane) is the union of the motorcycle traces of $m_1, \ldots, m_k$. Claim (1) states that the height of $\hat{m}_k$ is greater than or equal to the height of $\mathcal{T}$ at p. In order to prove this claim it suffices to show that the slope of $\mathcal{T}$ is at any interior point of a trace of $m_1, \ldots, m_k$ at most the slope of the tilted trace. The proof is an induction-type proof. We show (i) that $\mathcal{T}$ is convex within the interior of motorcycle traces, and (ii) that the slope constraint is maintained when migrating from one trace to the next.

(i) We start our considerations at $m_1(0)$. Due to the existence of m_1 there is a reflex wavefront vertex that started from $m_1(0)$. The wavefront vertex traces a reflex straight skeleton arc and has the same speed as m_1 by our definition of the motorcycles. Hence $\mathcal{T}$ and $\hat{m}_1$ start with the same slope. Let us consider the part T of $\mathcal{T}$ that lies above the trace of m_1. If there is a reflex

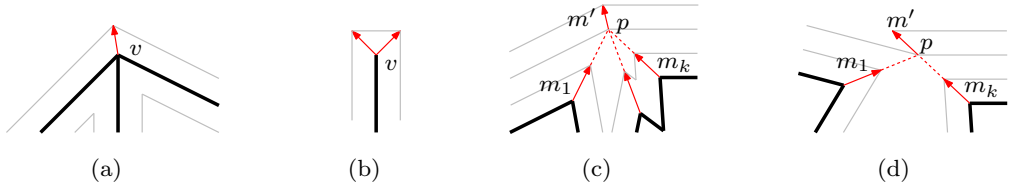

Figure 3: Four different situations where motorcycles are launched.

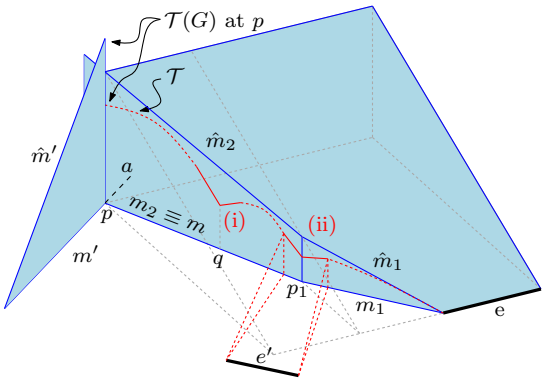

Figure 4: Proof of Theorem 2. The terrain $\mathcal{T}(G)$ is always below the tilted motorcycle traces $\hat{m}_k$. The proof shows that the slope of the terrain (red) is at any point at most the slope of the tilted motorcycle traces since the situations (i) and (ii) (shown in solid red and described in the proof) do not occur.

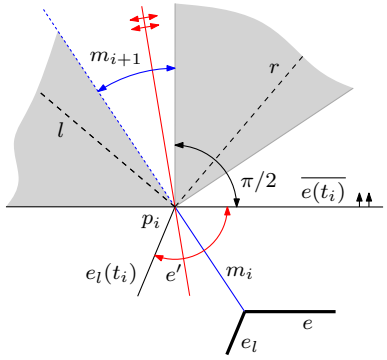

Figure 5: Proof of Theorem 2, case (ii). The shaded areas depict the valid domains for l and r and m_{i+1} is at any time right of l and left of r.

vertex in T we consider the one whose projection q on the plane is closest to $m_1(0)$. Obviously there would be a valley of $\mathcal{T}(G)$ at q. By assumption there would also be a motorcycle trace covering this valley at q. Since $\hat{m}_1$ is above (or just at the same height as) this trace it follows that m_1 would have crashed at q. This is a contradiction. Hence the slope of $\mathcal{T}$ is non-increasing above the trace of m_1. The same arguments suffice to show that $\mathcal{T}$ is non-increasing above the trace of m_i if $\mathcal{T}$ was below $\hat{m}_i$ at p_{i-1}.

(ii) Next we show that if the slope of $\mathcal{T}$ before p_i is less than or equal to the slope of $\hat{m}_i$, then the slope of $\mathcal{T}$ is less than or equal to the slope of $\hat{m}_{i+1}$ after p_i. We denote by e' the wavefront edge which defines $\mathcal{T}$ after p_i. The slope of $\mathcal{T}$ before (after, resp.) p_i can be expressed by the angle between the trace of m_i (m_{i+1}, resp.) and e'; the slope increases montonously as the corresponding angle increases. Hence, we can rephrase our assertion: if the angle between m_i and e' is smaller than the angle between m_i and e then the angle between m_{i+1} and e' is smaller than the angle between m_{i+1} and e. Let us consider Fig. 5. We denote by l the bisector between e and e' on the left side and by r the bisector on the right side. Hence, we have to prove that m_{i+1} lies right of l and left of r. Our premise states that the angle between e' and m_i is less than or equal to the angle between e and m_i. We denote by e_l the left arm of m_i and by t_i the time when e reaches p_i. Assume that we rotate e' counter-clockwise at p_i in Fig. 5, until e' is overlapping with $\overline{e(t_i)}$. Then l is falling onto $\overline{e(t_i)}$ and r is perpendicular to $\overline{e(t_i)}$. Vice versa, assume that we rotate e' clock-wise at p_i until e' is overlapping with e_l. Then l is on the supporting line of m_i and r is on the bisector of m_i and $\overline{e(t_i)}$. We shaded the valid domains for l and r in Fig. 5 accordingly. Since the angle between m_i and m_{i+1} is convex, m_{i+1} is right of the domain of l. Since m_{i+1} and e enclose an angle of at least $\pi/2$, m_{i+1} is left of the domain of r. Summarizing, for every position of e' which conforms to our initial assumption, m_{i+1} encloses a smaller angle with e' than with e. This concludes the assertion.

Combining arguments (i) and (ii) yields an induction-type proof for Claim (1). Basically, we showed that the distance between $\mathcal{T}$ and the tilted motorcycle traces is (not strictly) monotonically increasing. If $\mathcal{T}$ and the tilted motorcycle traces are overlapping until p then equality for the height of $\hat{m}$ and $\mathcal{T}(G)$ at p is attained. If at some point $\mathcal{T}$ leaves the tilted motorcycle traces then $\mathcal{T}(G)$ is strictly below $\hat{m}$ at p.

We now return our attention to Fig. 4 and use Claim (1) in a proof by contradiction of Thm. 2. So assume that there is a reflex arc a in $\mathcal{S}(G)$ which is only partially covered by a motorcycle trace m'. Hence, m' crashed into a motorcycle m. We denote by p the crashing point of m'. Without loss of generality we assume that the height of $\hat{m}'$ at p is lowest.[2] Hence, all valleys of $\mathcal{T}(G)$ are covered by motorcycle traces up to the height of $\hat{m}'$ at p. By Claim (1) we know that $\mathcal{T}(G)$ is below $\hat{m}$ at p. On the other hand we know that

[2]By this assumption we can assume that a is at least partially covered. If a would not be covered at all, then a is not incident to G and at least one of its reflex ancestor arcs is not covered completely.

$\mathcal{T}(G)$ has the same height as $\hat{m}'$ at p. (See the left side of Fig. 4.) This contradiction finally concludes the proof. □

Theorem 2 extends a result by Cheng and Vigneron [6]. They assumed that so-called vertex events do not happen and hence two motorcycles were not allowed to crash simultaneously. The idea of their proof is incorporated in Case (i) of Claim (1). This claim is a useful tool on its own grounds, and we cast it into the following corollary.

COROLLARY 3. *Let m be a motorcycle of $\mathcal{M}(G)$ and $p \in \mathbb{R}^2$ a point on the trace of m. The height of $\hat{m}$ is greater or equal to the height of $\mathcal{T}(G)$ at p. Equality is attained if and only if the valley of $\mathcal{T}(G)$, which corresponds to m, exists until p.*

LEMMA 4. *Let p, q be two distinct points on $\mathcal{T}(G)$. Then the slope of the line $\overline{pq}$ is at most 1.*

PROOF. We denote by p' and q' the projections of p and q onto the ground plane. Consider the intersection $\mathcal{T}$ of $\mathcal{T}(G)$ with a vertical curtain erected above the line section $[p', q']$. Then $\mathcal{T}$ is a planar monotone polygonal chain whose sections have a slope of at most 1. Hence the line $\overline{pq}$ has a slope of at most 1. □

Lower envelope. Let e denote a wavefront edge at time zero with endpoints a and b, see Fig. 6. If there is a motorcycle starting at a whose right arm is e then we consider the whole chain of tilted motorcycles traces, starting at a and ending at a', whose right arms are e. If there is no such motorcycle then $a' := a$. Analogously for the chain of tilted motorcycle traces starting at b and ending at b' whose left arms are e. Now we consider the plane slab which lies on the supporting plane of the terrain face of e and which is bounded below by e and the considered motorcycle traces, see Fig. 6. At the ends a' and b' the slab is bounded by rays which are perpendicular to $\overline{e(0)}$. Summarizing, at every input edge we have two slabs, one at each side, and for every terminal vertex we have one additional slab. We denote by $L(G)$ the lower envelope of the union of those slabs.

THEOREM 5. *The lower envelope $L(G)$ is identical to $\mathcal{T}(G)$.*

PROOF. It is easy to see that each face of $\mathcal{T}(G)$ is contained in its corresponding slab of $L(G)$. It remains to show that no point of $\mathcal{T}(G)$ is above $L(G)$. Assume to the contrary that a point $p \in \mathcal{T}(G)$ is above a slab of an edge e, see Fig. 6. We project p down to the slab and denote the projection point by u. Then we project u down along the steepest descent of the slab until we hit e or one of the tilted motorcycle traces and get the point v. If v is on a tilted trace then Cor. 3 implies that we can project v down to $\mathcal{T}(G)$ and get a point q. (Otherwise, $q := v$.) Since the line between u and v has slope 1, the slope of $\overline{pq}$ is greater than 1. This is a contradiction to Lem. 4. □

Computing $\mathcal{S}(G)$ using graphics hardware. Theorem 5 admits a simple method to render $\mathcal{T}(G)$ without knowing $\mathcal{S}(G)$. One first computes the motorcycle graph $\mathcal{M}(G)$ by a conventional algorithm (on the CPU) and then constructs the slabs as illustrated in Fig. 6. By rendering the set of slabs while looking from below at them one obtains an image which corresponds to $L(G)$. Hence, by employing techniques described by Hoff et al. [9], one can compute the straight skeleton of planar straight line graphs using graphics hardware.

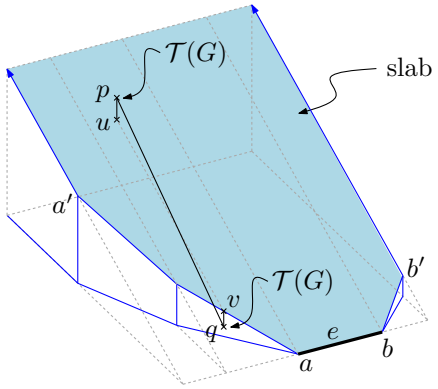

Figure 6: The shaded area depicts the slab defined by the wavefront edge e.

Theorem 5 extends the corresponding theorem of Cheng and Vigneron [6] who proved the claim for "non-degenerated simple polygons". Theorem 5 also extends a result by Eppstein and Erickson [7]. Instead of considering tilted motorcycle traces as the lower boundary of the slabs of Fig. 6, they considered the corresponding reflex arcs of $\mathcal{S}(G)$. Since the motorcycle traces cover the reflex arcs of $\mathcal{S}(G)$ it is easy to see that our slabs contain the slabs of [7]. This difference, however, is essential when attempting to compute $\mathcal{S}(G)$ via a lower-envelope computation. Besides, Thm. 5 provides an alternative, non-procedural way to define $\mathcal{S}(G)$ as the lower envelope of partial linear functions as in [7]. But, again, our definition of those functions does not depend on the lengths of the reflex arcs of $\mathcal{S}(G)$, but on the motorcycle graph instead.

3. A WAVEFRONT-TYPE ALGORITHM

Aichholzer et al. [1] presented a simple wavefront algorithm for straight skeletons of polygons. This algorithm basically simulates the propagating wavefront by processing the upcoming edge events and split events in chronological order. As the authors pointed out the hard problem is to efficiently determine the next split event. Our approach[3] is to avoid this problem by employing the motorcycle graph $\mathcal{M}(G)$ of the input graph G.

First, we extend the ordinary wavefront $\mathcal{W}(G, t)$ by adding the motorcycle graph $\mathcal{M}(G)$ as follows. We denote by $\mathcal{M}(G, t)$ those parts of $\mathcal{M}(G)$ which have not been swept by $\mathcal{W}(G, t')$ for $t' < t$. Then we insert $\mathcal{M}(G, t)$ into $\mathcal{W}(G, t)$ by splitting the edges of $\mathcal{W}(G, t)$ at the intersection points. Those intersection points will be called *moving Steiner vertices*. The other vertices of the graph $\mathcal{M}(G, t)$ correspond to crashing points of motorcycles. A vertex of $\mathcal{M}(G, t)$ that corresponds to a crashing point of two or more motorcycles will be called *multi Steiner vertex*. The remaining vertices of $\mathcal{M}(G, t)$ will be called *resting Steiner vertices*. The resulting graph is called the extended wavefront $\mathcal{W}^*(G, t)$, see Fig. 7. Again, we interpret $\mathcal{W}^*(G, t)$ as a kinetic planar straight-line graph.

[3]A preliminary version of this algorithm was described in [11].

LEMMA 6. *For any $t \geq 0$ the set $\mathbb{R}^2 \setminus \bigcup_{t' \in [0,t]} \mathcal{W}^*(G,t')$ consists of open convex faces.*

PROOF. This lemma follows immediately from Lem. 1 and the fact that every reflex angle of a reflex vertex of $\mathcal{W}(G,t)$ is split by (parts of) motorcycle traces of $\mathcal{M}(G,t)$. □

For $t = 0$, the lemma implies that $G + \mathcal{M}(G)$ induces a tessellation of the plane into convex faces. Another consequence of the lemma is that during the propagation of $\mathcal{W}^*(G,t)$ only vertices which are adjacent in $\mathcal{W}^*(G,t)$ can meet. In fact, this is one of the central aspects of our algorithm. A split event occurs when a reflex vertex meets the moving Steiner point which marks the end of the corresponding motorcycle trace. Thus, we avoid the costly search of the next split event.

The basic algorithm in order to compute $\mathcal{S}(G)$ is simple. First, for an arbitrarily small $\epsilon > 0$ we put every edge e of $\mathcal{W}^*(G,\epsilon)$ into a priority queue Q where the priority is given by the collapsing time of e (if it is finite). Next, we fetch the next event in Q, apply the corresponding topological change to $\mathcal{W}^*(G,t)$ and repeat until Q gets empty. The following event classes have to be distinguished, see Fig. 7:

- **(Classical) edge event** Two convex vertices u and v meet. We add the convex straight skeleton arcs traced out by u and v. Then we merge u and v to a new convex vertex. As a special case we check whether a whole triangle of the wavefront crashed due to u and v.
- **(Classical) split event** A reflex vertex u meets a moving Steiner vertex v and they are moving towards each other. First, we add a reflex straight skeleton arc which has been traced out by u. Then we consider the wavefront at the left of the edge $e = (u,v)$. If this side collapsed we add corresponding straight skeleton arcs. Otherwise a new convex vertex emerges, which is connected to the vertices adjacent to u and v lying left to e. We proceed likewise at the right side of e.
- **Start event** A reflex vertex or a moving Steiner vertex u meets a resting Steiner vertex v. So v becomes a moving Steiner vertex and one of the incident edges of u (but not (u,v)) is split by v.
- **Switch event** A convex vertex u meets a moving Steiner vertex or a reflex vertex v. The convex vertex u is migrating from one convex face to a neighboring one by jumping over v. If v was a reflex vertex then it becomes a moving Steiner vertex and we add corresponding straight skeleton arcs.
- **Multi split event** (a.k.a. vertex event) Reflex vertices $u_1, \ldots, u_k$ meet a multi Steiner vertex u. This event is conceptionally similar to the classical split event. We add reflex straight skeleton arcs which have been traced out by $u_1, \ldots, u_k$. Then we split the wavefront into multiple parts and insert convex and reflex vertices where necessary. (We skip the details due to lack of space.)
- **Multi start event** A moving Steiner vertex u meets a multi Steiner vertex v. We add a moving Steiner vertex for each edge that is incident to v (except for the edge (u,v)). The edges incident to u (except for (u,v)) are split by those new Steiner vertices and u itself is removed.
- **Remaining events** If two moving Steiner vertices meet then we can simply remove the corresponding edge. All other events (e.g. a convex vertex meets a resting Steiner vertex) are guaranteed not to occur.

Correctness. Our algorithm simulates the propagation of $\mathcal{W}^*(G,t)$ instead of $\mathcal{W}(G,t)$. The correctness follows from Thm. 2: reflex vertices of the wavefront do not leave $\mathcal{M}(G)$.

Runtime complexity. Let us assume that $\mathcal{M}(G)$ is given. The construction of the initial wavefront $\mathcal{W}^*(G,\epsilon)$ can easily be done in $O(n \log n)$ time. For each event we have to update Q for those edges which are new or where the collapsing time changed. A single modification takes $O(\log n)$ time. A single edge, split and start event requires $O(1)$ modifications and there are in total $\Theta(n)$ such events. Likewise, all multi split events and multi start events require in total $\Theta(n)$ modifications. A single switch event requires $O(1)$ modifications. Since a convex vertex does not meet a moving Steiner vertex twice the number of switch events is in $O(n^2)$. Moreover, since the number of Steiner vertices is bounded by the number r of reflex vertices we can refine the upper bound for the number k of switch events to $O(nr)$.

LEMMA 7. *If $\mathcal{M}(G)$ is given then our algorithm takes $O((n+k)\log n)$ time, where k is the number of switch events, with $k \in O(nr) \subset O(n^2)$.*

For practical applications it seems unlikely that more than $O(n)$ switch events actually occur and hence an actual runtime of $O(n \log n)$ may be expected in real world. Intensive experiments in the next section confirm this runtime complexity for virtually all of our tested datasets. However, a worst-case example for $O(n^2)$ switch-events can be constructed.

4. EXPERIMENTAL RESULTS

Our implementation Bone is implemented in C++ using ordinary double-precision floating-point arithmetic and the STL for standard data structures. The motorcycle graph is computed by our code Moca [10]. Moca is based on a geometric hash and achieves an $O(n \log n)$ runtime[4] in practice, provided that the motorcycles are sufficiently well distributed. (Some highly degenerate inputs may cause Moca to consume quadratic time, though.) In theory, more elaborate algorithms, with better worst-case complexities, by [7] and Cheng and Vigneron [6] are known. However, both algorithms seem far too complex to be implemented. Also, the algorithm by Cheng and Vigneron needs to know all motorcycles a-priori in order to compute the $1/\sqrt{n}$-cutting, making it difficult to apply it within our approach.

In the remainder of this section we report on a comparison of our implementation to the CGAL implementation[5] by Cacciola [5], which computes straight skeletons of simple polygons with holes. We used CGAL 3.7 in our tests. According to its documentation it is recommended to use CGAL's

[4]While experiments in [10] already confirmed this runtime characteristics, were able to improve the corresponding stochastic analysis. A full version of that paper is currently under review for publication.

[5]We use the term CGAL as a shorthand to refer to this implementation.

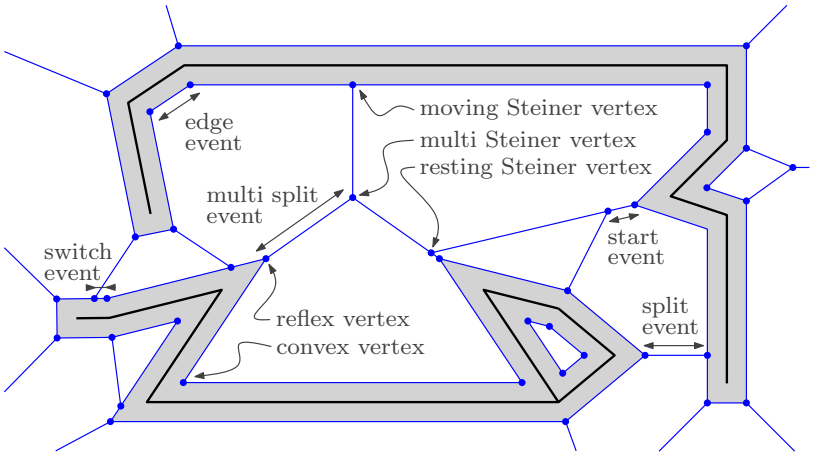

Figure 7: The extended wavefront $\mathcal{W}^*(G,t)$ is shown in blue. The shaded area has been swept by the wavefronts until time t.

exact-predicates-inexact-constructions kernel. In order to shed light on its basic runtime characteristics we also tested CGAL with inexact arithmetic.

We ran our experiments on a 64-bit Linux system with an Intel Core i5 processor clocked at 3.33 GHz and 8 Gb of memory. For practical reasons we limited the runtime to 10 minutes and the memory utilization to 6 Gb for the tested process. Our experiments were carried out on about 13 500 datasets of different characteristics, including (but not limited to) mechanical designs, printed circuit boards, freehand drawings, random polygons generated by RPG [4], font outlines, and different synthetic datasets like fractals. Since CGAL supports only polygons with holes we restrict our experiments to such datasets. In order to measure the performance of CGAL we computed the interior straight skeleton of the polygon with holes, the interior straight skeletons of the holes and the exterior straight skeleton of the outer face up to the offset distance 100 (after scaling the input to fit into the unit square).

In Fig. 8 we plot the runtime of (a) Bone, (b) Bone without the time consumed by the computation of $\mathcal{M}(G)$, (c) CGAL using exact predicates and inexact constructions, (d) CGAL using inexact arithmetic. For the vast majority of datasets Bone consumes $c \cdot n \log n\, \mu s$, where n denotes the number of vertices in the input graph G and $10 \leq c \leq 30$. Figure 8(b) indicates that virtually all of the outliers are due to the computational efforts for computing the motorcycle graph $\mathcal{M}(G)$. CGAL supposedly has a worst-case complexity of $O(n^2)$ and Fig. 8 (c) shows that it needs at least $1.7 \cdot n^2 \log n\, \mu s$ time. Interestingly, the runtime of CGAL for datasets of the same size n varies within two orders of magnitude. Fig. 8 (d) suggests that this variation in the runtime is mainly due to the exact-arithmetic kernel. That is, the exact-arithmetic kernel leads to a performance drop by roughly a factor 3 to 100. Still, even with inexact arithmetic CGAL seems to consume at least $\Omega(n^2)$ time.

We note that Bone handles datasets with more than 10^6 vertices while CGAL was not able to process polygons with more than about 10^4 vertices within the time and memory limits imposed. Interestingly, even when run with inexact arithmetic, 6 Gb as memory limit did not allow CGAL to progress significantly beyond 10^4 vertices. For the comparatively small datasets CGAL could be run on, the difference in the practical ("average") complexities of Bone and CGAL results in Bone being about one to two orders of magnitude faster than CGAL.

Reliability tests of Bone on several thousands of additional data sets confirmed the practical suitability of our approach even for complex data. Still, Bone-internal sanity checks or inspections of the wavefront patterns revealed a few numerical problems, and we are working on fine-tuning the handling of multiple nearly-parallel wavefronts in order to boost Bone to industrial strength.

5. CONCLUSION

A generalization of motorcycle graphs to arbitrary PSLGs G allows us to extend important results by Cheng and Vigneron [6] and Eppstein and Erickson [7]: the motorcycle graph $\mathcal{M}(G)$ is guaranteed to cover all reflex arcs of the straight skeleton $\mathcal{S}(G)$ even if vertex events are allowed. Furthermore, the lower envelope of slabs erected above $\mathcal{M}(G)$ allows to obtain $\mathcal{S}(G)$ via a projection to the ground plane. As an immediate application of this non-procedural description of $\mathcal{S}(G)$ we sketch a simple algorithm for rendering $\mathcal{S}(G)$ on a graphics hardware. The relation between $\mathcal{M}(G)$ and $\mathcal{S}(G)$ is exploited in a novel wavefront-type algorithm for computing straight skeletons of PSLGs. Our new algorithm is easy to implement and runs in $O(n \log n)$ time in practice (if $\mathcal{M}(G)$ is given). Extensive experiments revealed an average runtime of $20 \cdot n \log n\, \mu s$ of our implementation Bone (for the computation of $\mathcal{M}(G)$ and $\mathcal{S}(G)$) on a standard PC for virtually all of our 13 500 datasets. A comparison with the straight-skeleton implementation provided by CGAL revealed a speed-up by a factor of n, irrelevant of whether CGAL was run with or without exact arithmetic. Our algorithm could be easily tweaked to compute the linear axis [12] of planar straight-line graphs, since the linear axis is based on the same wavefront propagation process. Besides further improving the speed and reliability of Bone when run with conventional floating-point arithmetic, we plan to interface it with an EGC library and to extend the underlying algorithm to weighted straight skeletons.

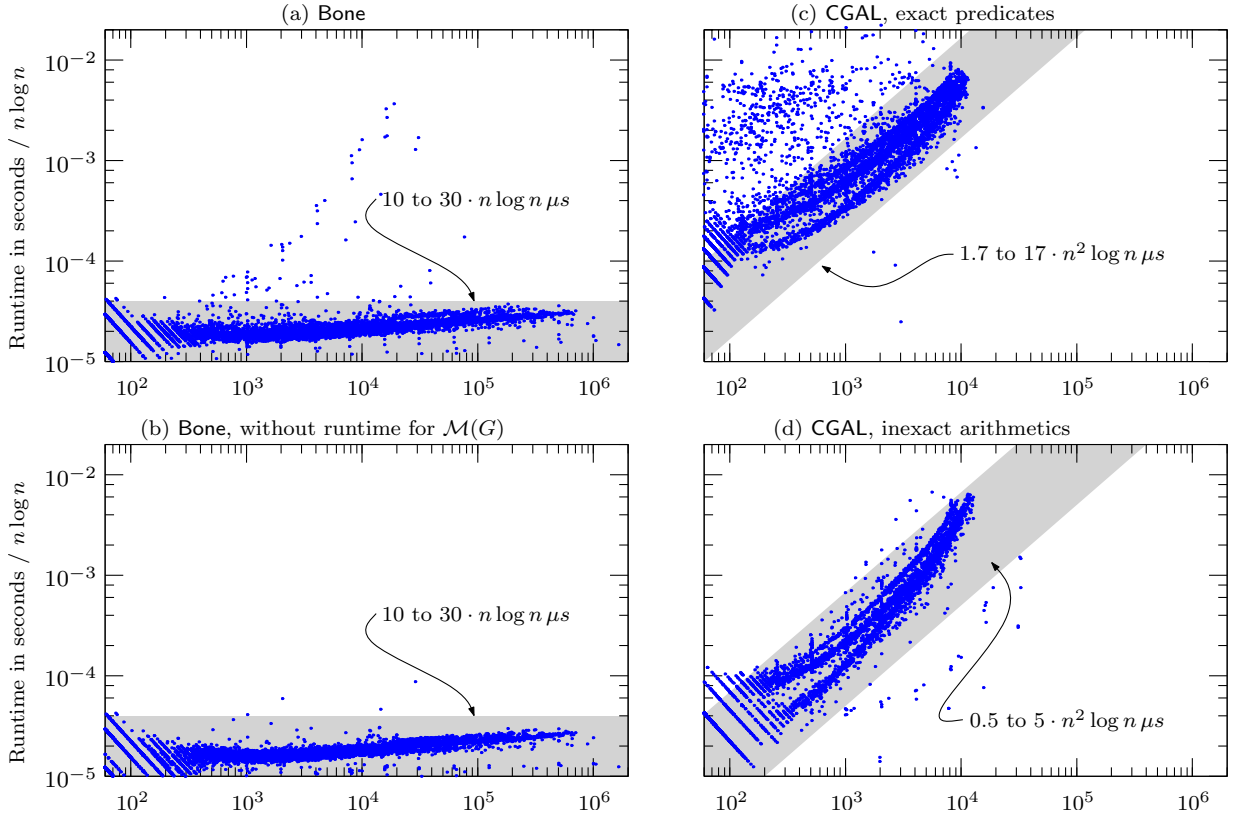

Figure 8: Every point depicts the runtime for a single dataset. The x-axis denotes the number n of vertices of the input, ranging from 60 to $2 \cdot 10^6$. For illustrative reasons we divided the actual runtime in seconds by $n \log n$. Hence, a horizontal line corresponds to an $n \log n$ complexity. Points in the shaded areas correspond to runtimes as labeled.

6. REFERENCES

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

[2] O. Aichholzer and F. Aurenhammer. Straight Skeletons for General Polygonal Figures. In *Proc. 2nd Annu. Internat. Conf. Comput. Combinatorics*, volume 1090 of *Lecture Notes Comput. Sci.*, pages 117–126. Springer-Verlag, 1996.

[3] O. Aichholzer and F. Aurenhammer. Straight Skeletons for General Polygonal Figures in the Plane. In A. Samoilenko, editor, *Voronoi's Impact on Modern Science, Book 2*, pages 7–21. Institute of Mathematics of the National Academy of Sciences of Ukraine, Kiev, Ukraine, 1998.

[4] T. Auer and M. Held. Heuristics for the Generation of Random Polygons. In *Proc. Canad. Conf. Comput. Geom. (CCCG'96)*, pages 38–44, Ottawa, Canada, Aug 1996. Carleton University Press.

[5] F. Cacciola. A CGAL Implementation of the Straight Skeleton of a Simple 2D Polygon with Holes. In *2nd CGAL User Workshop*, Polytechnic Univ., Brooklyn, New York, USA, June 2004.

[6] S.-W. Cheng and A. Vigneron. Motorcycle Graphs and Straight Skeletons. *Algorithmica*, 47:159–182, Feb 2007.

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

[8] P. Felkel and Š. Obdržálek. Straight Skeleton Implementation. In *Proc. 14th Spring Conf. Comput. Graphics*, pages 210–218, Budmerice, Slovakia, 1998.

[9] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha. Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware. In *Comput. Graphics (SIGGRAPH '99 Proc.)*, pages 277–286, Los Angeles, CA, Aug 1999.

[10] S. Huber and M. Held. A Practice-Minded Approach to Computing Motorcycle Graphs. In *Proc. 25th Europ. Workshop Comput. Geom.*, pages 305–308, Brussels, Belgium, Mar 2009.

[11] S. Huber and M. Held. Computing Straight Skeletons of Planar Straight-Line Graphs Based on Motorcycle Graphs. In *Proc. 22nd Canad. Conf. Comput. Geom. (CCCG 2010)*, pages 187–190, Winnipeg, Canada, Aug 2010.

[12] K. Vyatkina. Linear axis for planar straight line graphs. In *Proceedings of the Fifteenth Australasian Symposium on Computing: The Australasian Theory - Volume 94*, CATS '09, pages 139–152, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc.

APPENDIX

A. APPENDIX

A.1 Proof of Lemma 1

Consider a point p of $\mathcal{M}(G)$ which does not coincide with G. Then a local disc around p is tessellated into convex slices by $\mathcal{M}(G)$.

PROOF. If p is in the relative interior of a trace this assertion is trivially true. Hence, we assume that p is the endpoint of $k \geq 2$ motorcycle traces, and denote the corresponding motorcycles by $m_1, \ldots, m_k$. Consider a local disc D around p. If D is tessellated into convex slices by the traces of $m_1, \ldots, m_k$ then our assertion holds. So assume that there is a non-convex slice. W.l.o.g., assume that the motorcycles are numbered in a cyclic order such that (i) their corresponding traces appear counter-clockwise around p and (ii) the trace of m_1 and m_k bound the reflex slice.

If the left arm of m_1 and the right arm of m_k span a convex angle then there is a motorcycle m which continues the movement of m_k and the assertion holds again. So, assume that the left arm of m_1 and the right arm of m_k span a reflex angle. Hence there is a motorcycle m which started from p and shares its left arm with m_1 and its right arm with m_k. We have to prove that the traces of m_1 and m as well as m_k and m span a convex angle.

Without loss of generality we assume that m and m_k span an angle greater than π. The basic idea is to prove that under this assumption the existence of m is contradicted because (i) m_1 or m_k crashes before reaching p or (ii) m_1 and m_k reach p after a fourth motorcycle was at p before.

We denote by e_l and e_r the left and right arm of m, respectively. We note that m shares its right arm with m_k and its left arm with m_1. Denote by e' the right arm of m_1 and by e the left arm of m_k, see Fig. 9 (a). Note that neither m_1 nor m_k need to start at time zero. We denote by R the right-most ancestor chain of motorcycle traces of m_1, i.e., the sequence of motorcycle traces from m_1 till the wavefront segment $e'(0)$. Likewise, we denote by L the left-most ancestor chain of motorcycle traces of m_k.

First, it is easy to see that $e(0)$ is above $\overline{e_l(0)}$, where "above $\overline{e_l(0)}$" means being on the propagation side of $\overline{e_l(0)}$: The point $m_1(0)$ must lie on $\overline{e_l(0)}$. (If m_1 does not start at time zero we mean by $m_1(0)$ the extrapolation of its movement before its actual starting time.) Likewise, $m_k(0)$ must lie on $\overline{e_r(0)}$. By assumption, $m_k(0)$ lies left of the speed ray of m and right of the speed ray of m_1. Hence the arms of m_k span a smaller angle (on the propagation side of the motorcycle) than the one of of m, which means that $e(0)$ lies above $\overline{e_l(0)}$.

We denote by Δ the triangle enclosed by $\overline{e_l(0)}$ and the supporting lines of the traces of m_1 and m_k. Among all vertices of G within Δ we consider a vertex v that has maximum orthogonal distance to $\overline{e_l(0)}$. Note that at least that endpoint of $e(0)$ that is not incident to L is within Δ. We denote by n the propagation vector of e_l and by l the parallel line of e_l through v, see Fig. 9 (a). If there are multiple vertices of G which lie on l then we may assume that v is the left-most on l (that is, the closest to the trace of m_1).

Consider l being moving parallel with speed vector n. There is always a motorcycle m' emanating from v which is at any time in front of the moving line l (or just coinciding). In order to see that we can distinguish two cases regarding the number of incident edges of v:

- The vertex v has two or more edges incident. Since no incident edge of v lies above l we have a motorcycle m' emanating from v, see Fig. 9 (b). We denote by e'' the left arm of m'. Note that $e''(0)$ is not collinear with l; otherwise there would be another vertex of G on l which is left to v.

  We denote by s the bisector of l and e'' which consists exactly of those points which are reached by the propagating $\overline{e''}$ and l at the same time. Every point right of s is reached by $\overline{e''}$ at first. Since the motorcycle m' is at any time right of s we see that m' is always in front of the moving line l.
- The vertex v is a terminal vertex. There are two motorcylces emanating from v. We denote by m' the one whose speed vector has the greater inner product with n, see Fig. 9 (c). We denote by e'' the arm of m' which is parallel to the incident edge of v. (W.l.o.g. we may assume that this is the left arm.)

  As in the first case we denote by s the bisector of the moving line l an e''. Since m' is right of s (or just overlapping) we see that m' is never behind the moving line l.

To sum up, there is always a motorcycle m' which never falls behind the moving line l. Next, we note that l intersects R and L in their relative interior. We distinguish the following two cases:

Case 1 Assume that m' reaches R or L: It is easy to see that if we move l parallel by the speed vector n then l is always in front of $\overline{e_l(t)}$. That means that the intersection of l with R (resp. L) is always in front of the motorcycle m_1 (resp. m_k) and its right-most (resp. left-most) ancestor motorcycles. (All points right of the (top-down) bisector line between l and $\overline{e(0)}$ are visited by l first.) Since m' moves w.r.t direction n at least as fast as l we see that m' crosses R or L and hence (i) m_1 or one of its ancestors crashes into m', or (ii) m_k or one of its ancestors crashes into m' or (iii) m_1 and m_k reach p but m' was there before. In any case m is not emanated. This yields a contradiction.

Case 2 Assume that m' does not reach R or L: Hence, m' crashed within Δ. The motorcycle m'', in which m' crashed, must have started below l, since no motorcycle could have come through R or L. Hence m'' is faster than m' w.r.t. direction n. We consider m'' as the new m' and apply again our case analysis. Since there is only a finite number of motorcycles we eventually end up in Case 1 and get a contradiction.

□

Computing the motorcycle graph. The motorcycle graph problem was introduced by Eppstein and Erickson [7], drawing its motivation from the computation of straight skeletons. They presented an algorithm that runs in $O(n^{17/11+\epsilon})$ time and space. Their original formulation of the motorcycle graph problem did not include motorcycles that start when two or more other motorcycles crash simultaneously. Still, since their algorithm is based on closest-pair queries on a dynamic set of objects which model the traces

and moving motorcycles, it could be applied to our extended version of the motorcycle graph problem. However, their algorithm is too complex to be implemented such that the theoretical runtime complexity is matched indeed.

Cheng and Vigneron [6] introduced an $O(n\sqrt{n}\log n)$ algorithm based on $1/\sqrt{n}$-cuttings. Unfortunately, this algorithm actually needs to know all motorcycles a-priori in order to compute the $1/\sqrt{n}$-cutting. For actual implementations the motorcycle graph $\mathcal{M}(G)$ could be computed in $O(n^2 \log n)$ by a priority-queue enhanced brute-force algorithm, cf. [6].

For our straight skeleton implementation we use our motorcycle graph implementation Moca [10]. It is based on a geometric hash and provides an $O(n \log n)$ runtime[6] in practice, provided that the motorcycles are sufficiently well distributed.

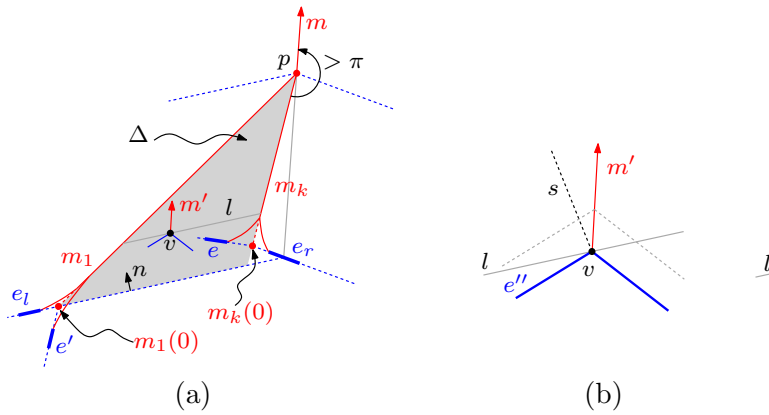

Figure 9: Convex tessellation at simultaneous crashes. (a) If m and m_k span a reflex angle then the launch of m is impossible. (b–c) A close-up of the vertex v, where v is (b) a non-terminal vertex and (c) a terminal vertex.

[6]While experiments in [10] already confirmed this runtime behavior, we were able to improve the corresponding stochastic analysis. A full version of that paper is currently under review for publication.