Straight Skeletons and their Relation to Triangulations*

Stefan Huber† Martin Held†

Abstract

We study straight skeletons of polygons and investigate the dependence of the number of flip events of the classical wavefront propagation by Aichholzer and Aurenhammer on the underlying triangulation. We show that their standard algorithm, applied to a polygon with n vertices, has to cope with at least $\Omega(n^2)$ flip events. In particular, $\Omega(n)$ diagonals of a triangulation may reappear $\Omega(n)$ times each. Still, by allowing a linear number of Steiner points in the triangulation we can avoid flip events completely. As an application of this result we explain how the straight skeleton of a simple polygon with n vertices can be computed in time $O(n^2 \log n)$ by a wavefront-based algorithm that matches the simplicity of the algorithm by Aichholzer and Aurenhammer.

1 Introduction

Since the introduction of straight skeletons by Aichholzer et al. [2] many questions related to straight skeletons have remained unanswered. In particular, there is a significant gap between the known lower and upper bounds for computing straight skeletons: While the best lower bound for the computation of the straight skeleton of a simple polygon with n vertices is $\Omega(n \log n)$, the fastest known algorithms by Eppstein and Erickson [4] and Cheng and Vigneron [3] provide a worst-case runtime of $O(n^{17/11+\epsilon})$ and an expected runtime of $O(n^{3/2} \log n)$, respectively. Both algorithms seem very difficult to implement and, in fact, no implementation is known.

In terms of implementability an approach based on triangulations by Aichholzer and Aurenhammer [1] looks much more promising. They take a triangulation of the input polygon[1] and simulate a wavefront propagation by bookkeeping topological changes of the underlying moving triangulation. Edge and split events of the straight skeleton correspond to topological changes in the triangulation. However, additional topological changes of the triangulation — so-called flip events — appear when a vertex crosses a triangulation diagonal. The handling of the $O(n)$ split and edge events costs $O(n^2 \log n)$ time, while a single flip event can be processed in $O(\log n)$. It is widely believed (but yet unproven) that the number of flip events is bounded by $O(n^2)$, which would yield an overall $O(n^2 \log n)$ bound for their algorithm.

In Sec. 2 and 3 of this paper we present results related to the unproven $O(n^2)$ bound on the number of flip events. Subsequently, in Sec. 4 we use Steiner points to obtain triangulations of polygons that are free of flip events. As an application of this result we explain in Sec. 5 how the straight skeleton of a simple polygon with n vertices can be computed in time $O(n^2 \log n)$ by a wavefront-based algorithm that matches the simplicity of the algorithm by Aichholzer and Aurenhammer [1].

2 How often can diagonals reappear?

Currently, the best known upper bound for the number of flip events is $O(n^3)$. This follows from the fact that three points that move with constant speed along lines in the plane are either never, once, twice or always collinear. (The determinant of the matrix whose columns consist of the homogeneous coordinates of the points is a quadratic expression in time, and a root indicates collinearity.) Consequently, a single diagonal of a triangulation can be crossed at most $2(n-2)$ times by other vertices, and since there are at most $O(n^2)$ possible diagonals, an upper bound of $O(n^3)$ follows. Of course, not every collinearity of three vertices corresponds to a flip event.

This proof links the number of flip events with the number of reappearances of triangulation diagonals. The following lemma highlights that one cannot establish an $O(n^2)$ bound on the number of flip events during the wavefront propagation by attempting to show that the number of reappearances of every single diagonal of an arbitrary triangulation is in $O(1)$.

Lemma 1 *There exists a sequence of polygons P_n, with $\Theta(n)$ vertices, and corresponding triangulations T_n such that $\Omega(n)$ diagonals of T_n reappear $\Omega(n)$ times during the wavefront propagation applied to P_n.*

Proof. Roughly, we come up with an appropriate geometric configuration of moving vertices that realizes a sequence of topological transitions such that diagonals reappear as often as claimed. Our constructive proof is split into three parts. First, we construct a polygon P and a triangulation T where one diagonal

*Work supported by Austrian FWF Grant L367-N15.

†Universität Salzburg, FB Computerwissenschaften, A-5020 Salzburg, Austria, {shuber,held}@cosy.sbg.ac.at

[1]Actually, their algorithm can handle planar straight line graphs as input.

reappears twice. Then we extend this construction such that a single diagonal appears $\Omega(n)$ times. In the third part we extend it such that $\Omega(n)$ diagonals each reappear $\Omega(n)$ times. (Due to the lack of space we omit details of Part 3 of the proof, though.) In the sequel we denote by $V(t)$ the position of the vertex V at time t and by AB the supporting line of the vertices A and B.

Part 1: We start with showing how to make a diagonal AB reappear twice during the movement of six vertices A, B, S_1, S_2, N_1, N_2 as induced by the wavefront propagation. The wavefront propagation starts at time $-\varepsilon$, for a sufficiently small $\varepsilon > 0$, and topological transitions will occur at times $0, 1, 2, 3, 4$ and 5. The initial positions of the six vertices (discussed in detail below) are shown in the top part of Fig. 1, while the upper-left triangulation in the lower part of Fig. 1 shows the initial triangulation of the vertices. The other triangulations show topological transitions needed to recreate AB. Roughly, AB will disappear because the vertex S_1 crosses it. Then S_1 falls back behind AB again. After the recreation of that diagonal $S_2(t)$ will cross it, causing it to disappear again. The corresponding topological transitions are illustrated in the bottom of Fig. 1.

We now discuss details of the geometric configuration of the six vertices. Let the two vertices A, B both

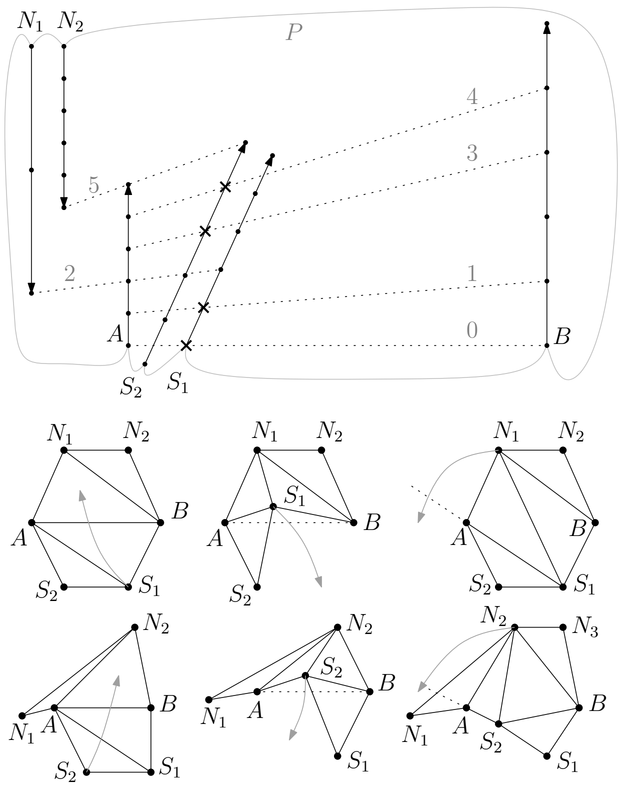

Figure 1: Part 1. Top: geometric configuration. Bottom: Topological transitions at the six points in time depicted as grey numbers in the top sub-figure.

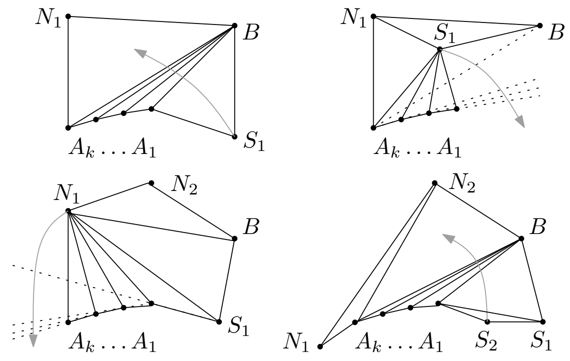

Figure 2: Part 3. Topological transitions of the first reappearance cycle.

move northwards (parallel to the positive y-axis), with B moving strictly faster than A. We want the vertex S_1 to move northeastwards and to cross AB at time 0 and to fall back behind AB a bit later, say at time 1. We achieve this by demanding $S_1(t)$ to lie on $A(t)B(t)$ for $t \in \{0, 1\}$, cf. Fig. 1. Once $S_1(0)$ and $S_1(1)$ are fixed, the movement of S_1 has been specified completely. Since the area of the triangle $\Delta(A(t)B(t)S_1(t))$ is a quadratic expression in t and since $A(t)B(t)S_1(t)$ are oriented clockwise (CW) for some $t > 1$ we conclude that $A(t)B(t)S_1(t)$ is oriented CW for $t \notin [0, 1]$ and counter-clockwise (CCW) for $t \in (0, 1)$. The recreation of the diagonal AB is achieved by letting a vertex N_1 move southwards to the left of A such that it crosses $A(t)S_1(t)$ at, say, time 2, cf. Fig. 1. Now we place a vertex S_2 that moves parallel to S_1 between A and S_1 such that $S_2(t) \in A(t)B(t)$ for $t \in \{3, 4\}$. Again it holds that $A(t)B(t)S_2(t)$ is oriented CW if $t \notin [3, 4]$ and CCW if $t \in (3, 4)$. Thus, the diagonal AB disappears at time 3 and S_2 falls back behind AB again at time 4. Similar to N_1 we place a vertex N_2 that recreates the diagonal AB by requesting $N_2(5) \in A(5)S_2(5)$, cf. Fig. 1. Note that by increasing the inclination of the supporting rays of the vertices S_1, S_2 we can force the start positions of $S_1(-\varepsilon)$ and $S_2(-\varepsilon)$ to get arbitrarily close to the line $A(-\varepsilon)B(-\varepsilon)$. By doing so we can guarantee that a polygon P exists such that the given geometric configuration is achieved. (It is shown as a curve depicted in light grey in Fig. 1.)

Part 2: We add vertices $S_3, \ldots, S_m$ from right to left between S_2 and A, according to the construction scheme of part 1. That means that S_{i+1} gets collinear with A and B at time $3i$ and $3i + 1$. Analogously we add vertices $N_3, \ldots, N_m$ from left to right next to N_2 such that N_{i+1} crosses AS_{i+1} at time $3i + 2$. Again, note that if the vertices $S_1, \ldots, S_m$ are moving nearly vertically then the start positions of $S_1, \ldots S_m$ get arbitrarily close to the line AB.

Part 3: The basic idea is to arrange copies $A_2, \ldots, A_k$ of A_1 along a reflex chain of the polygon such that the positions of $A_2(t), \dots, A_k(t)$ remain sufficiently close to the line $A_1(t)B(t)$ for the entire time span of the wavefront propagation. By doing so one can achieve the topological transitions as sketched by the first reappearance cycle in Fig. 2. □

3 Can we always find good triangulations?

As a byproduct of the previous lemma we obtain polygons and triangulations that lead to $\Omega(n^2)$ flip events. However, this result hinges on meticulously chosen triangulations which are bad in the sense that $\Omega(n)$ diagonals each reappear $\Omega(n)$ times. For example, in the construction scheme above, we could have initially put diagonals between N_m and $A_1, \dots, A_k$, thus avoiding the reappearance cycles of the diagonals. Can we always find, for every given polygon P, a good triangulation T such that the number of flip events is low, say $o(n^2)$ or even $O(n)$? The following lemma provides a negative answer to this question.

Lemma 2 *There exist polygons with n vertices for which every triangulation leads to $\Omega(n^2)$ flip events.*

Proof. We consider the polygon shown in Fig. 3. The vertices $A, E_1, \dots, E_k, B$ lie on a reflex chain such that W is the only vertex initially seen by an E_i. Hence every triangulation contains the diagonals WE_i, for $1 \le i \le k$. These diagonals are flipped by the notch vertices $N_1, \dots, N_m$ which move southwards. We ensure that $N_1, \dots, N_m$ are fast enough that they cross those diagonals before any edge or split event occurs where a vertex $E_1, \dots, E_k$ is involved. Furthermore, we request that for each $i \in \{1, \dots, m-1\}$ the vertex N_{i+1} does not cross the supporting line of AE_1 before N_i induced $\Omega(k)$ flip events. The claim follows by choosing m, k roughly equal to $n/2$. □

One might feel that retriangulating at specific favorable moments could reduce the complexity. However, the polygon above seems to illustrate a counter example: Note that a retriangulation at any point in time does not save more than $O(n)$ flip events. Hence, to gain a runtime advantage one would have to perform $\Omega(n)$ retriangulations. Assume that we perform a retriangulation saving $O(n)$ flip events. Handling the flip events directly would have cost $O(n \log n)$ time. A single retriangulation invalidates $O(n)$ entries in the priority queue. This means that one would gain a runtime advantage only if $O(n)$ entries in the priority queue could be reset in $o(n \log n)$ time.

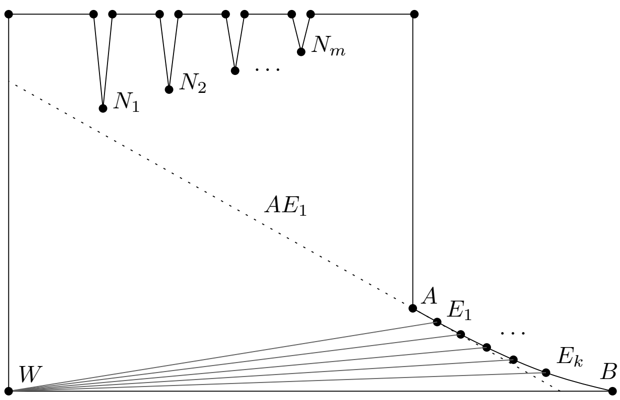

Figure 3: A polygon where every possible triangulation leads to $\Omega(n^2)$ flip events.

4 Steiner triangulations without flip events

In this section we investigate whether Steiner points can be used to obtain triangulations with a low number of flip events or, more precisely, with no flip events at all.

Lemma 3 *Every simple polygon P with n vertices admits a triangulation that employs at most $n-2$ Steiner points and which is free of flip events.*

Proof. We consider the straight skeleton S of P and add its at most $n-2$ inner nodes as Steiner points and its arcs as initial diagonals of the triangulation. It now remains to triangulate the faces of S.

Let the face f of S be induced by the segment s of P. Let p, q denote the endpoints of s. Note that f (as a polygon) is monotone with respect to s and that reflex vertices of f are only present in the corresponding monotone chain that does not contain s. Recall that split event nodes correspond to reflex vertices of f. If f contains no reflex vertices then we triangulate f arbitrarily. Otherwise we choose that reflex vertex v which has minimum orthogonal distance to s, and insert two diagonals vp and vq. Note that f contains the diagonals completely: otherwise we would have missed a split event node of f having smaller orthogonal distance to s. Then f is decomposed by the triangle pqv into two remaining parts A and B, where A contains the diagonal pv and B contains the diagonal qv, see Fig. 4. We proceed recursively within A and B. That is, if A has no reflex vertex then we triangulate arbitrarily. Otherwise we find a reflex vertex v' with minimum orthogonal distance to s and insert two diagonals $v'p$ and $v'v$. Then A is split by the triangle pvv' into two parts, and so on.

During the wave propagation the vertices of P move on the straight skeleton and hence do not cross any diagonal at any time. For every face f the corresponding segment s is moving in a self-parallel manner inwards and may be split when reaching reflex vertices of f. In contrast to that the Steiner points stay in place and wait until the corresponding segments of P reach them. The triangles in a face collapse only when s reaches a node of f and hence an edge or a split event occurs. However, no diagonal crosses a Steiner point such that a flip event needs to be handled. □

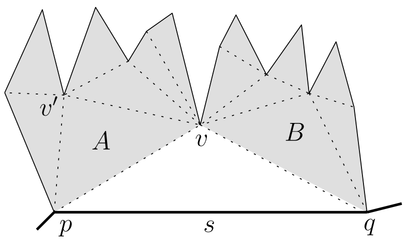

Figure 4: The triangulation scheme for a straight skeleton face of a segment s.

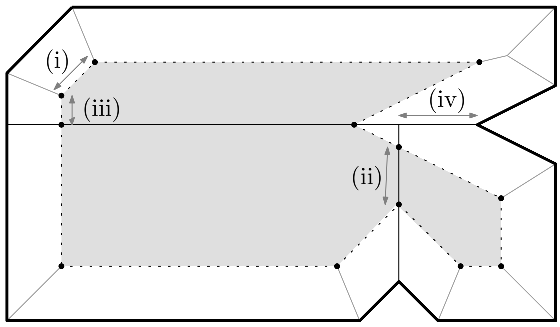

Figure 5: Different types of events for the motorcycle graph based straight skeleton algorithm.

5 Motorcycle graph based straight skeletons

Obviously, the proof of the previous lemma does not result in a new algorithm for computing straight skeletons. However, we recall again that convex vertices of P do not cause flip events. On the other hand, in the former construction, reflex vertices of P are barred from causing flip events since those vertices move along triangulation diagonals which are part of the straight skeleton. Fortunately, this property also holds if we replace the straight skeleton by the motorcycle graph M induced by the moving reflex vertices.

For the sake of simplicity we also adopt the assumption of Cheng and Vigneron [3] that no split event of higher degree exists, i.e., that no two or more reflex vertices meet simultaneously in a common point. Under this assumption Cheng and Vigneron [3] showed that a reflex arc of the straight skeleton is not longer than the trace of the corresponding motorcycle. (We assume that motorcycles crash at the boundary of the initial polygon.) Note that M always decomposes P into convex parts during the entire shrinking process.

This suggests that we can obtain an algorithm for computing straight skeletons by by employing the motorcycle graph during the wavefront propagation process. In contrast to the former sections we do not consider a triangulation but maintain the intersection points of M with the wave front and call them Steiner vertices. The following types of events occur, see Fig. 5: (i) edge event: two neighboring convex vertices in a convex part of P meet; (ii) split event: a reflex vertex meets its corresponding Steiner vertex; (iii) switch event: a convex vertex meets a Steiner event and hence the convex vertex migrates to a different convex part of P; (iv) start event: a reflex vertex or a (moving) Steiner vertex meets a (resting) Steiner vertex, which is the endpoint of a different trace and has to start moving. Note that only neighboring[2] vertices meet in the propagation process since the motorcycle graph decomposes the shrinking polygon P at any time into convex parts.

Hence, it suffices to check only for collisions among vertices which are neighbors. We put corresponding events into a priority queue and process them in chronological order. In the worst case there are up to $O(n^2)$ switch events, but all other events occur $\Theta(n)$ times. Every event can be handled in $O(\log n)$ time, since only a constant number of neighbors of the two vertices affected have to be modified in their propagation speed. Hence, the algorithm runs in $O(n^2 \log n)$ time in the worst case but still enjoys a simplicity that is comparable to the triangulation-based algorithm by Aichholzer and Aurenhammer [1].

We note that the $O(n^2)$ bound on the number of switch events seems overly pessimistic, and there is reason to assume that for many data sets, in particular for those from real-world applications, there might be only $O(n)$ switch events, resulting in $O(n \log n)$ runtime in practice (if the motorcycle graph is already given). Furthermore, our algorithm need not be restricted the interior of one polygon but could also be extended to planar straight-line graphs.

[2] On the wave front or on the motorcycle graph.

Acknowledgement We would like to thank Gerhard Mitterlechner for valuable discussions.

References

[1] 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.

[2] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761, 1995.

[3] S.-W. Cheng and A. Vigneron. Motorcycle Graphs and Straight Skeletons. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 156–165, San Francisco, CA, USA, 2002.

[4] 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.