

Straight Skeletons of Monotone Surfaces in Three-Space

Martin Held*

Peter Palfrader*

Abstract

We present a simple algorithm to compute the straight skeleton and mitered offset surfaces of a polyhedral terrain in 3D. Like its 2D pedant, the 3D straight skeleton is the result of a wavefront propagation process, which we simulate in order to construct the skeleton in time $\mathcal{O}(n^4 \log n)$, where n is the number of vertices of the terrain. Any mitered offset surface can then be obtained from the skeleton in time linear in the combinatorial size of the skeleton.

1 Introduction

The straight skeleton was introduced to computational geometry over 20 years ago by Aichholzer et al. [2]. Let P be a simple polygon in the plane and consider the following process. At time $t = 0$, each edge of P starts to move towards the interior of P at unit speed in a self-parallel manner, maintaining incidences. The set of moving edges forms a set of polygons $\mathcal{W}_P(t)$, called the wavefront of P at time t . Note that each edge of $\mathcal{W}_P(t)$ is at all times at orthogonal distance t to its corresponding edge of P .

The wavefront needs to be updated at times to remain a set of simple polygons: As edges shrink to zero length (*edge event*), they are removed, and edges are split and incidences updated when a previously non-incident vertex moves into their interior (*split event*). (If the polygon is not in general position then more complex interactions are possible.) The straight skeleton $\mathcal{S}(P)$ of P is then defined as the geometric graph whose edges consist of the traces of wavefront vertices over the propagation process, see Figure 1.

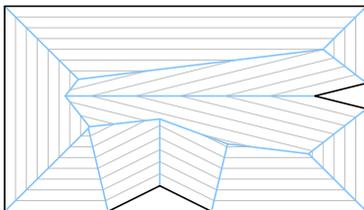


Figure 1: The straight skeleton $\mathcal{S}(P)$ (blue) of an input polygon P (bold) is the union of the traces of the vertices of P as it shrinks. Several instances (wavefronts) of the shrinking polygon are shown in gray.

*Universität Salzburg, FB Computerwissenschaften, 5020 Salzburg, Austria; supported by Austrian Science Fund (FWF) Grant P25816-N15; {held,palfrader}@cosy.sbg.ac.at.

A mitered offset of P at offsetting distance t corresponds to the wavefront at time t . Mitered offsets are inherently linked to the straight skeleton: Given the straight skeleton $\mathcal{S}(P)$ of a polygon P with n vertices, any mitered offset can be constructed in $\mathcal{O}(n)$ time and space [11].

Variations of the straight skeleton problem in the plane have also been investigated, by generalizing the input to arbitrary planar straight line graphs or by adding multiplicative or additive weights to input edges [1, 7, 8, 10].

The algorithm with the currently best worst-case complexity for computing the 2D straight skeleton of an arbitrary simple polygon is due to Eppstein and Erickson [8] and requires both $\mathcal{O}(n^{17/11+\varepsilon})$ time and space, for an arbitrarily small but positive ε . Better runtime bounds can be obtained when restricting the input to monotone polygons. Indeed, Biedl et al. [6] present a simple and easy-to-understand algorithm which requires $\mathcal{O}(n \log n)$ time and linear space to compute the straight skeleton.

Moving to 3-space. Straight skeletons of polytopes were studied by Barequet et al. [5], and recently by Aurenhammer and Walzl [4]. However, while combinatorial complexities have been established for the straight skeleton of polytopes, no runtime bounds have been investigated.

In this work, we consider the straight skeleton and mitered offsets of polyhedral terrains in 3-space. As usual, a polyhedral terrain is a piecewise linear, continuous function of two variables. To simplify matters, we assume that the terrain \mathcal{T} is defined over all of \mathbb{R}^2 and that all facets are simply-connected. Furthermore, we assume \mathcal{T} is in general position: No more than four supporting planes of the facets of \mathcal{T} shall be tangent to a common sphere and the degree of any vertex of \mathcal{T} shall be at most a constant k .

2 Wavefront Propagation

We consider the wavefront propagation of a polyhedral terrain \mathcal{T} . Just as in the plane, where the 2D-wavefront consists of edges at distance t to their corresponding input edge, here the wavefront consists of wavefront facets which are at orthogonal distance t to their corresponding input facets at all times.

Formally, let f be a facet of \mathcal{T} , let \bar{H}_f be its supporting plane, and let \vec{n}_f be the unit normal of f with

positive z -coordinate. Then we define the offset supporting plane at distance t to be $H_f(t) := \bar{H}_f + t \cdot \bar{n}_f$. The wavefront, just like the input \mathcal{T} , is a continuous, piecewise linear surface, i.e., a polyhedral terrain. Its facets at time t are embedded in the offset supporting planes $H_f(t)$ of all facets f of \mathcal{T} .

Initially, at time $t = 0$, the wavefront $\mathcal{W}_{\mathcal{T}}(t)$ is identical to \mathcal{T} . When the propagation process starts, all facets of the wavefront move upwards, in positive z -direction. During this propagation, incidences are retained where possible.

For the initial offset at time $t = \delta$, for a sufficiently small $\delta > 0$, retaining the combinatorial structure is possible along edges. Furthermore, locally at vertices of degree three, an offset of the same combinatorial structure is possible. However, at vertices of degree four or more, any offset, even at an infinitesimally small δ , will generally have a combinatorial structure different from the input: The offset surface consists locally of several degree-three vertices that arise from the offsets of the planes incident at the input vertex of higher degree; see Aurenhammer and Walzl [4].

2.1 Events

As the wavefront propagation continues, the combinatorial structure of the surface has to be updated and the set of wavefront vertices and their trajectories change at discrete points in time at so-called *events*, when four or more wavefront facets pass through a common point.

Aurenhammer and Walzl [3] consider straight skeletons of polytopes, and they differentiate between events that change the topology of the offset polytope and events that merely change the surface of the polytope. They call the first class *solid events*, which includes *splitting events*, where the polytope disconnects and *piercing events*, where a vertex runs into a facet. However, since our wavefront surface is z -monotone and continuous, these events cannot occur and we will only observe the second class of events, *surface events*, in the wavefront propagation.

An *edge event* happens at time t when an edge of the wavefront collapses to zero length without its incident facets vanishing, too. The two vertices incident at the edge are merged, giving rise to a high-degree vertex. For the wavefront after the event, at time $t + \delta$, this high-degree vertex has to be resolved and generally split again similar to the process at the initial wavefront construction. See Figure 2a.

A second type of event, the (*facet*) *split event* happens when a vertex v of the wavefront that is incident at facet f moves into the interior of another edge e of f without f collapsing. This case is similar to the split event known from 2D straight skeletons. Combinatorially, the edge e is split at the locus of v and made incident to v , creating a higher-degree vertex

which then needs to be resolved again for the post-event wavefront. See Figure 2b.

In the third event type, the *face event*, a facet f may collapse to an empty area. This coincides with one or more edges of f collapsing or a vertex of f moving into the interior of another edge of f . At the event time t , the facet is replaced by a set of edges that cover its boundary without overlapping, thereby merging vertices which now occupy the same locus (if such vertices exist). Again for the post-event wavefront, higher-degree vertices may need to be resolved. See Figure 2c.

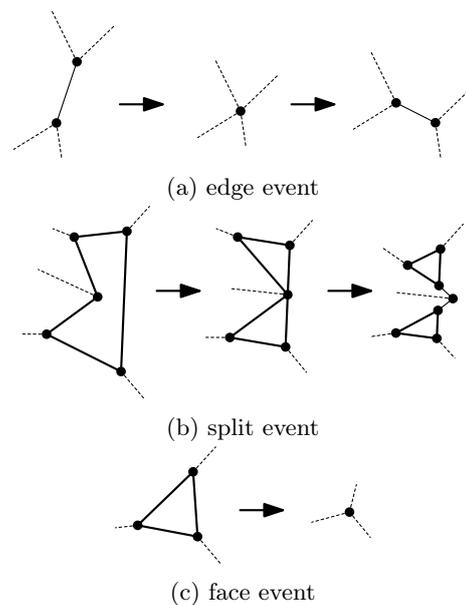


Figure 2: Edge event, split event, and face event during the wavefront propagation.

If the input is not in general position, two vertices that share a facet but not an edge can also meet. This will result in a higher-degree vertex (usually of at least degree six) that needs to be split again. We have ruled out such cases by our general position assumption.

2.2 Computing the Straight Skeleton

Once no more events occur during the propagation process, the process has finished. The three dimensional straight skeleton $\mathcal{S}(\mathcal{T})$ of \mathcal{T} is then the structure whose edges are the traces of wavefront vertices and whose facets are the traces of wavefront edges. To unambiguously refer to features of the 3D straight skeleton, Aurenhammer et al. [4] call the edges of $\mathcal{S}(\mathcal{T})$ *spokes* and its facets *sheets*. The volumes bounded by sheets are called *cells*.

Interior vertices correspond to events that have been observed in the propagation process. Any wavefront vertex or edge remaining in the wavefront at the end of the process induces an unbounded straight skeleton spoke or sheet which continues to infinity.

3 Simulating the Wavefront Propagation

We compute the straight skeleton $\mathcal{S}(\mathcal{T})$ of \mathcal{T} by simulating its wavefront propagation. This requires determining at every stage in the process what the next event will be. To cope with this problem we maintain a priority queue of potential events: As initialization, we first create the initial wavefront for time $t = \delta$, where δ is infinitesimally small, splitting higher degree vertices of \mathcal{T} . Then we store for every edge of the wavefront its collapse time, and we store for every vertex of the wavefront the instances of when it will move into any of the edges of its incident facets.

To advance time in our simulation of the wavefront propagation, we fetch the event from the priority queue with the earliest associated time. We process this event by modifying the wavefront combinatorics according to the event type, thereby merging and then splitting vertices as required and as described in the previous section. We add new events to the priority queue for all edges and vertices that were affected or created by the event.

Then, we proceed and fetch the next item from the priority queue. We need to verify that it still is a valid event, that is, we need to check that the edge that is supposed to collapse or the vertex that is supposed to move into an edge are still elements of the wavefront — prior events may have already restructured the wavefront and invalidated this event. If it is a valid event then we process it as described. Otherwise we simply drop it. In either case, this process is repeated until the priority queue is empty.

Number of events. In general, an event happens at point p and time t when four (or more) wavefront facets become incident. (For simplicity reasons, our general position assumption guarantees that no more than four wavefront facets are involved in an event.) This provides a natural upper bound of $\binom{n}{4}$ on the size of the priority queue, where n is the number of facets of the input surface. Based on our experience with different algorithms for computing straight skeletons in the plane, we conjecture that in practice only a small subset of those $\binom{n}{4}$ combinations will be relevant.

Splitting higher-degree vertices. Aurenhammer and Walzl [3] note that an offset surface of a higher-degree vertex v of a three dimensional polytope always exists even though is not necessarily unique. One offset that always exists corresponds to a wavefront where v has been replaced by a tree. In [4], they suggest as a simple approach to enumerate all combinatorially different trees and check whether they correspond to valid offset surfaces of v . The geometry of a tree's element is dictated by its combinatoric properties. Such a valid tree will replace the vertex v in the propagating wavefront.

By our general position assumption, all vertices of the input surface have at most constant degree k . Thus, finding this tree for a single vertex v is a constant-time operation as well. Furthermore, at most a constant number of elements need to be added to the wavefront per input vertex.

Vertex degrees during events. After having constructed the initial wavefront, all moving vertices will be of degree three in the generic case. We investigate the types of vertices that can appear in events.

In an edge event, the edge that connects to degree-three vertices collapses, giving rise to a degree-four vertex v , as shown in Figure 2a. In the generic case, v will have to be split (at constant cost) into two new degree-three vertices connected by a new edge. In our general position assumption we stated that no more than four supporting planes of faces may be tangent to a common sphere. Thus, for our input we will always either split v into two, or v will never again participate in an event.

In a split event, a degree-three vertex v comes to lie on previously non-incident wavefront edge e , which is split in two during the event, giving rise to a degree-five vertex (Figure 2b). In the generic case this vertex will be split into three new vertices, each of degree three. Again, by our general position assumption, this will be the case for our input sets.

For face events we can distinguish two sub-types (Figure 3). In one, a triangle facet will collapse as all its incident edges shrink to zero length. This will give immediate rise to a new degree-three vertex which can then propagate. The other type is where a more complex polygon collapses as some of its edges collapse and maybe some vertices become incident at other edges of the polygon. The facet is replaced by one or more edges, and all resulting vertices will be of degree three and can propagate without any need to be split. Note that multiple face collapses happening at the same time may cause an edge that has the same face on both sides. Such an edge is not removed; instead it propagates like any other edge, similar to how ghost vertices propagate in Biedl et al. [7]. This ensures that all faces remain simply connected during the propagation.

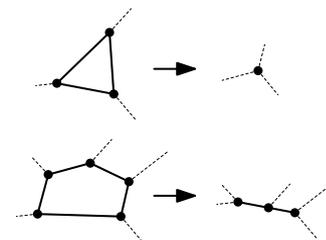


Figure 3: Two types of face collapsing events.

4 Obtaining Offset Surfaces

If only a single mitered offset surface at orthogonal distance t is sought, then one approach to construct this offset is to simply run the wavefront propagation process until time t . Then, the wavefront at this time is the offset surface required.

However, if multiple offset surfaces at different distances should be constructed or if the straight skeleton is already available, then we apply the following process to obtain an offset surface in time linear in the size of the straight skeleton:

For a given spoke s , we denote by $s(t)$ the three dimensional point obtained by intersecting s with a plane at distance t and parallel to the base of any one of its incident cells. Equivalently, $s(t)$ is the location at time t of the wavefront vertex that traced out s .

For every spoke s of the skeleton which exists at (orthogonal) offsetting distance t , i.e., for which $s(t)$ exists, and for every cell c incident at s where (s, c) has not been processed before, we construct an offset facet as follows: Let f_1 be one of the sheets of $s_1 := s$ that is on the boundary of c . We walk along the boundary of f_1 , moving in the direction of positive z , until we reach another spoke s_2 of f_1 that exists at distance t . Now let f_2 be the other sheet of c incident at s_2 and repeat the walk in f_2 to find a spoke s_3 . Eventually, we will return to our initial spoke s_1 . Let s_ℓ be the last one before we returned. (Special handling will be required to process the case of infinite elements.) The polygon with vertices $s_1(t), s_2(t), \dots, s_\ell(t)$ is now a valid offset facet and we add it to the offset surface we are constructing. We then mark $(s_1, c), (s_2, c), \dots, (s_\ell, c)$ as processed and continue with our main loop. Once all spokes have been processed, the set of offset facets represents the complete offset surface. This algorithm can be implemented such that all offset facets together with their adjacency relations are obtained.

The correctness of this approach hinges on the property that all offset facets are simple polygons and contain no holes. This property stems from the fact that the wavefront propagation does not experience any piercing event since \mathcal{T} is a terrain.

5 Discussion

We have presented a simple algorithm to compute the straight skeleton of a z -monotone surface. The processing cost of each event is constant for generic input, and the number of events is bounded by $\binom{n}{4}$. We do not expect this bound to be tight, though. Maintaining the events in a priority queue results in a runtime bound of $\mathcal{O}(n^4 \log n)$. Better upper bounds are currently under investigation. A construction by Held [9] establishes an $\Omega(n^2)$ lower bound on the combinatorial complexity of $\mathcal{S}(\mathcal{T})$ for a terrain \mathcal{T} . His construction can be adapted to yield the same bound for the

combinatorial complexity of one mitered offset.

For descriptive simplicity, our general-position assumption bounds the maximum degree of a vertex that may appear in the propagating wavefront by a small constant. However, using larger constants does not change the process significantly and only results in more complex event handling requirements.

Furthermore, we can relax the bound on the maximum degree of vertices of the input surface. Resolving higher-degree vertices where the degrees are not bound by a constant for the initial wavefront will require more than constant work, but at least for pointed vertices, where all incident faces are confined to one half space, offsetting can be reduced to computing weighted 2D straight skeletons [5] which are well studied [7] and for which implementations exist [10, 11]. Vertices that are saddle-points can still be handled by one of the methods described by Aurenhammer and Walzl [4].

References

- [1] O. Aichholzer and F. Aurenhammer. Straight Skeletons for General Polygonal Figures in the Plane. In *Voronoi's Impact on Modern Sciences II*, volume 21, pages 7–21. 1998.
- [2] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A Novel Type of Skeleton for Polygons. *J. Univ. Comp. Sci*, 1(12):752–761, 1995.
- [3] F. Aurenhammer and G. Walzl. Three-dimensional straight skeletons from bisector graphs. In *5th Int. Conf. Analytic Number Theory & Spatial Tessellations*, 2013.
- [4] F. Aurenhammer and G. Walzl. Straight Skeletons and Mitered Offsets of Nonconvex Polytopes. *DCG*, 56(3):743–801, 2016.
- [5] G. Barequet, D. Eppstein, M. T. Goodrich, and A. Vaxman. Straight Skeletons of Three-Dimensional Polyhedra. In *ESA 2008*, pages 148–160.
- [6] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader. A Simple Algorithm for Computing Positively Weighted Straight Skeletons of Monotone Polygons. *IPL*, 115(2):243–247, 2015.
- [7] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader. Weighted Straight Skeletons in the Plane. *CGTA*, 48(2):120–133, 2015.
- [8] D. Eppstein and J. Erickson. Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions. *DCG*, 22(4):569–592, 1999.
- [9] M. Held. On Computing Voronoi Diagrams of Convex Polyhedra by Means of Wavefront Propagation. In *CCCG 1994*, pages 128–133.
- [10] M. Held and P. Palfrader. Additive Weights for Straight Skeletons. In *EuroCG 2016*.
- [11] P. Palfrader and M. Held. Computing Mitered Offset Curves Based on Straight Skeletons. *Comp.-Aided Design & Appl.*, 12(4):414–424, Feb. 2015.