THREE-DIMENSIONAL STRAIGHT SKELETONS FROM BISECTOR GRAPHS

FRANZ AURENHAMMER AND GERNOT WALZL

ABSTRACT. A *straight skeleton* of a polygon or of a polytope is a
piecewise linear skeletal structure that partitions the underlying ob-
ject by means of a self-parallel shrinking process. We propose a
method for constructing different straight skeletons for a given non-
convex polytope $\mathcal{Q}$ in 3-space. The approach is based on so-called bi-
sector graphs on the sphere, and allows for generating straight skele-
tons with certain optimality properties.

The various events that arise during the process of shrinking $\mathcal{Q}$
are discussed. We have implemented our method and give some ex-
amples of the output.

1. INTRODUCTION

Let P be a simple polygon in the plane. A skeletal structure for P
is a geometric graph that reflects the boundary structure of P, and
thus its (approximate) shape, in a combinatorially and computation-
ally useful manner. The most prominent representative is a Voronoi
diagram-like structure [5], the *medial axis* of P. It consists of all points
inside P that have more than one nearest neighbor on P's boundary; see
e.g. [4, 11]. The medial axis is a tree composed of line segments and par-
abolic segments. More recently, piecewise linear skeletons have received
increased attention. Among them are the *triangulation axis* of P, a con-
cept based on certain (optimal) triangulations of the polygon P, see [2],
and the *straight skeleton* of P, which is composed of angular bisectors
of P [1, 3, 8, 12].

We are interested in the last structure in the present paper. A well-
known procedural definition exists for the straight skeleton, by a self-
parallel offsetting process (shrinking process) for P and the resulting
'events' that construct the skeleton nodes. Events are unique changes
in the polygon boundary, yielding the *mitered offset* of P, in contradis-
tinction to the Minkowski sum offset specified by the medial axis of P.

Applications exist in diverse areas, including computer graphics, robotics, architecture, and geographical information systems; see e.g. [5] and the references above.

When generalizing to 3 dimensions, the construction of skeletal structures gets surprisingly involved, even for the medial axis of a nonconvex polytope $\mathcal{Q}$. To construct a straight skeleton for a given polytope $\mathcal{Q}$ in 3-space, its boundary facets are offset in a self-parallel fashion. Thereby, $\mathcal{Q}$ undergoes changes of geometric, combinatorial, and topological nature. Geometrical changes, of course, take place continuously, whereas combinatorial changes (in $\mathcal{Q}$'s boundary structure) and topological changes (like new tunnels, or breaking the polytope apart) occur once in a while. Each type of change implies the former ones. During the shrinking process, the edges and vertices of the polytope trace out the facets and edges, respectively, of its 3-dimensional straight skeleton.

Unlike parallel offsets of polygons, however, parallel offsets of polytopes in $\mathbb{R}^3$ are in general not unique. This makes the analysis and computation of 3-dimensional straight skeletons difficult. The main problem lies in offsetting the polytope vertices of degree $m \geq 4$. Such vertices may already be part of the input polytope $\mathcal{Q}$, and they necessarily arise during the shrinking process for $\mathcal{Q}$.

Not much literature exists on that topic. If $\mathcal{Q}$ is a *convex* polytope then its straight skeleton coincides with the medial axis of $\mathcal{Q}$ and can be computed in $O(n^2)$ time. Similarly, if $\mathcal{Q}$ is an *orthogonal* (i.e., axes-aligned) polytope then the straight skeleton is the medial axis of the polytope for the L_∞-metric. In both cases, we obtain a unique structure which, in contrast to general straight skeletons, can be defined via distances. Barequet et al. [7] and Martinez et al. [9] studied the orthogonal case and gave respective construction algorithms. The former work also considers the general case, and a systematic treatment is given in Aurenhammer and Walzl [6].

Here we propose an alternative method for computing straight skeletons of general nonconvex polytopes, which is more friendly to implementation. Also, the new approach is capable of generating all possible solutions, and can be used to construct 3D straight skeletons with predefined optimality properties.

2. Bisector graphs

We define a *polytope $\mathcal{Q}$* in 3-space $\mathbb{R}^3$ as a bounded, closed, and interior-connected subset of $\mathbb{R}^3$ with piecewise linear boundary. The boundary components of $\mathcal{Q}$ are faces of dimensions 2, 1, and 0. They are called
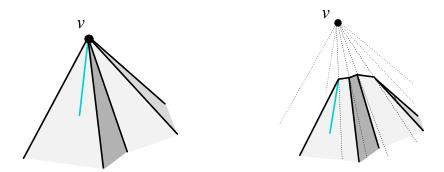
FIGURE 1. The degree-6 vertex v splits into four vertices of degree 3 in the local offset surface.

facets, *edges*, and *vertices* of $\mathcal{Q}$, respectively. A polytope is, in general, nonconvex and may contain tunnels, and even holes that make its boundary disconnected. However, if the input polytope $\mathcal{Q}$ is boundary-connected, no holes can be created in its offsetting process. In the easiest case, $\mathcal{Q}$ is homeomorphic to a ball in $\mathbb{R}^3$.

Let now v be some vertex of $\mathcal{Q}$. If v is of degree 3 only, then the boundary of $\mathcal{Q}$ will not change combinatorially in the neighborhood of v when the polytope shrinks slightly. Otherwise, there will be changes in the non-degenerate case, which split v into several vertices of degree 3, as is shown in Figure 1. We describe these changes now, with the help of so-called bisector graphs.

Let e be an edge of $\mathcal{Q}$ which is incident to vertex v. We consider the two facets f_i and f_j adjacent in e, and their supporting planes H_i and H_j. When $\mathcal{Q}$ is shrinking, the edge e moves on an angular bisector plane, B_{ij}, of H_i and H_j, because these planes will offset at unit speed towards the interior of $\mathcal{Q}$. Therefore, each local offset surface at v will have its edges contained in such bisector planes. The vertices of the offset surface move on trisector lines $t_{ijk} = B_{ij} \cap B_{ik} \cap B_{jk}$, as these planes intersect three by three in a straight line. Note that all such bisector planes and trisector lines pass through the vertex v. They are the supporting planes (respectively, lines) of the 3D straight skeleton components to be constructed.

As mentioned earlier, local offset surfaces need not be unique. We call such a surface *valid* if it is homeomorphic to a disk. To describe all valid solutions, we intersect the planes B_{ij} with a sphere, U, centered at v. We denote the resulting great circles by $b_{ij} = B_{ij} \cap U$. The system $(b_{ij})_{1 \le i < j \le m}$ defines an arrangement of great circles on the sphere U. Here $m \ge 4$ denotes the degree of v.
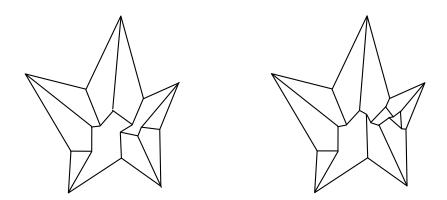
FIGURE 2. Two out of five valid bisector graphs that arise from a polytope vertex of degree 10. The spherical polygon $\mathcal{P}$ is an (almost flat) pentagonal star. By combining such graphs into a valid bisector graph for a vertex of high degree m, we obtain $\Omega(2^m)$ valid solutions.

For each valid offset surface, its edges will project radially to certain parts of this arrangement. Let us assume that U is small enough to intersect only faces of $\mathcal{Q}$ that are incident to v. Then $\mathcal{Q} \cap U$ is a simply connected spherical polygon, $\mathcal{P}$, whose boundary is a closed Jordan curve. *Bisector graphs* G that correspond to valid offset surfaces can now be defined inside the polygon $\mathcal{P}$.

G is defined as a graph on U with labelled arcs $a_{ij} \subseteq b_{ij}$, where the ordering of the labels (ij) indicates the position of the polytope facets f_i and f_j with respect to the bisector plane B_{ij}. G contains nodes of degree 1 (the nodes of $\mathcal{P}$), and of degree 3 whose incident arcs have labels of the form (ij), (ik), (kj). G is called *valid* if it is a crossing-free graph inside $\mathcal{P}$.

Lemma 1. *Every valid bisector graph G for the system $(b_{ij})_{1 \leq i < j \leq m}$ lifts, by central projection respect to v, to the edge graph of a valid local offset surface $\Gamma(G)$ for v.*

Proof. We lift G's arcs a_{ij} to the lines $\ell_{ij} = H_i^\varepsilon \cap H_j^\varepsilon$ by central projection with respect to v. (Superscript ε means parallel offset toward the interior of $\mathcal{Q}$, by an amount of $\varepsilon > 0$.) This lifting can be done because a_{ij}, ℓ_{ij}, and v are contained in the same plane, B_{ij}. Each connected face on U defined by G then lifts to a polygon in 3-space, which is planar because its edges e_{ij} are labelled with the same offset plane index j on the 'inside', by the labelling of G. That is, G lifts to a unique piecewise linear surface $\Gamma(G)$. The facets of $\Gamma(G)$ fit continuously, because their edges e_{ij} are part of ℓ_{ij} and thus lie in both offset planes H_i^ε and H_j^ε.

$\Gamma(G)$ is radially monotone with respect to v, because G is crossing-free. Hence $\Gamma(G)$ is a valid local offset surface. □

G may be disconnected and contain cycles. In fact, there may exists exponentially many valid bisector graphs, in the degree m of v. Figure 2 sketches an example.

3. Finding valid graphs

Lemma 1 implies that local offset surfaces can be computed by finding bisector graphs. All possible bisector graph for a spherical polygon $\mathcal{P}$ are contained in the corresponding arrangement the great circles b_{ij} define on the sphere U.

Let G be any bisector graph for the system (b_{ij}). We consider the surface defined by G when the translated facet planes H_i^ε are 'put in' according to G's labelling. For fixed $\varepsilon > 0$, this surface, $\Gamma(G)$, is unique. Now, if G is not valid then $\Gamma(G)$ will have self-intersections. In particular, certain parts of $\Gamma(G)$ will be expanding instead of shrinking when ε is increased. However, self-intersections define edges that project to arcs on U that can be used to form another bisector graph, that is, we can remove such expanding parts from $\Gamma(G)$ and get a valid surface.

If the resulting surface, $\Gamma(G')$, contains cycles then these can be removed as well. Each cycle corresponds to a single facet of $\Gamma(G')$, and when removing the respective plane H_i^ε we obtain a surface without this cycle. This leads to a cycle-free and valid bisector graph G'', which is a forest with $\mathcal{P}$'s nodes as its leaves.

The reason why G'' can be disconnected is that more than one polytope facet may be defined by the same plane H_i. We can complete G'' to a tree by introducing an 'edge' on each such facet of $\Gamma(G'')$ that makes G'' disconnected. In conclusion, we have:

Lemma 2. *There always exists a valid bisector graph for the system (b_{ij}) that is a tree.*

Algorithmically, in order to find a valid offset surface, it suffices to enumerate all unrooted binary trees with m leaves. Several such trees might be valid, which gives freedom in choosing the surface for further shrinking.

All such trees can be easily generated combinatorially; see e.g. [10]. For each tree, we then check if it has a crossing-free embedding as a bisector graph within $\mathcal{P}$. The number of different trees is exponential (a Catalan number), but usually the degree m is a small constant, independent of the number of polytope vertices. This is because most solids can be
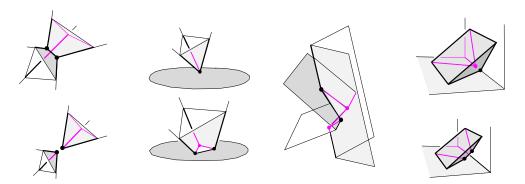
FIGURE 3. Solid events (from left to right): Splitting event, piercing event, kissing event, and (one possible) lifting event. The fifth solid event, the tetrahedra collapse, is not shown.

accurately approximated by (boundary-triangulated) polytopes having vertices of small constant degree. Also, in the non-degenerate case, the shrinking process leads to vertices of degree at most 8, as we shall see in Section 4.

4. Event handling

An *event* is a change in the combinatorial structure of the boundary of the shrinking polytope. It is caused by four or more offsetting facet planes that pass through a common point. Events that also change the topology of the polytope will be called *solid events*. If only the boundary of the polytope is affected but the topology stays the same, we talk of *surface events*.

Solid events are easier to understand and handle, at least in the non-degenerate case. Consult Figure 3. In a so-called *splitting event*, the polytope breaks apart: either globally, such that there are two interior-connected components now, which we shrink separately; or locally, when a tunnel gets destroyed. New tunnels may occur in several different ways: In a *piercing event* where a polytope vertex runs into a facet, in a *kissing event* where two polytope edges run into each other, and in a *lifting event* where two faces touch and then lift above the surface. The chronologically last event in the shrinking process is a solid event too, namely, the collapse of the polytope into a point which vanishes afterwards (a so-called *tetrahedra collapse*).

This list of solid events is complete. There is no other possibility for two faces of the polytope to get in contact with each other in a solid event, unless parallelism occurs. However, we have to include for each event its *inverse* event, too. (Such inverse events occur if we exchange
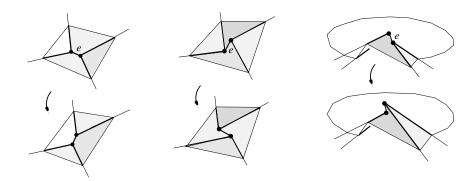
FIGURE 4. Three surface events (of type vertex touch).

the interior of the polytope with its exterior, and circumscribe it with a suitable bounding box.) Note that the lifting event and the inverse splitting event are similar; two polytope vertices get in touch. There is a unique way to proceed in the shrinking process after each solid event. We have therefore implemented these events directly, without resorting to bisector graphs.

Surface events are less easy to categorize, and their anatomy is less obvious. They arise initially (and simultaneously) if the input polytope contains vertices of degree higher than 3, and also later during the shrinking process. Initially, vertices of (potentially) arbitrary degree might have to be resolved. In later events, the degree will be bounded by 8 in the non-degenerate case, as we will see, but coplanarities and collinearities may arise generically, which complicates matters again. However, all surface events can be handled uniformly with the bisector graph approach in Sections 2 and 3, because for such events the polytope intersects the sphere U in a single spherical polygon. Below we concentrate on surface events that arise after the initial ones.

We distinguish between events of the types *vertex touch, vertex/edge touch*, and *edge/edge touch*. Again, this categorization is complete if inverse events are included, because only faces of dimensions 0 and 1 can get in touch in a surface event. (In fact, all events but the vertex/edge touch turn out to be self-inverse.)

A vertex touch may take place in several different ways. The simplest case is when a polytope edge, e, shrinks to length zero, thereby creating a vertex v of degree 4. Consult Figure 4. On the left side, all involved polytope edges are convex. (An edge e of a polytope $\mathcal{Q}$ is called *convex* if there exists no line segment $\ell \subset \mathcal{Q}$ whose interior intersects the interior of e in a single point. Otherwise, edge e is called *reflex*.) Edge e 'flips'
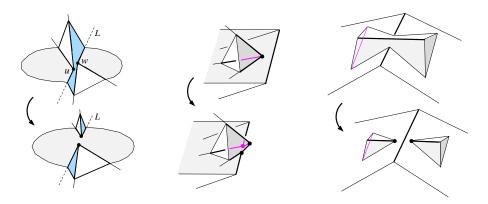
FIGURE 5. Degree-6 vertex touch, vertex/edge touch, and edge/edge touch.

into another convex edge in a unique way. In the middle picture, the created vertex v is a saddle point; the four edges incident to v alternate in being convex and reflex. Edge e now either flips and changes the convexity type, or it reappears and keeps the adjacency between the dark-shaded facets. Both choices lead to a valid offset surface after the event. Adjacency between facets is also kept in the right picture, where a saddle point v is created whose incident four edges do not positively span $\mathbb{R}^3$. The horizontal facet and the dark-shaded facet stay neighbors. This event is unique again. In any of these events, the degree-4 vertex v splits into two degree-3 vertices in one or the other way.

Note that simultaneous occurrences of such events will necessarily take place. For example, three simultaneous vertex touch events cause the collapse of a triangular polytope facet. For the sake of clarity, we do not detail such combined events here.

A vertex touch event does not have to include a collapse of a polytope edge, as Figure 5 (left) shows. The coincidence of the two approaching polytope vertices u and w that touch in v is not by degeneracy but by construction; they move on the intersection line L of two facet planes. Vertex v is of degree 6 in this case, and splits into two vertices of degree 3. Thereby the dark-shaded facet splits into two facets, and the two coplanar horizontal facets merge into one, on which two tetrahedral peaks drift apart.

A vertex/edge touch is depicted in Figure 5 (middle). The flat wedge retracts to the left in the shrinking process, faster than does the tetrahedral pyramid on its top. At v, the pyramid splits the rim of the wedge, whose lower facet then expands to above the rim in the offset polytope.

A degree-5 vertex v arises intermediately, which splits into three vertices of degree 3.

Figure 5 (right) shows an edge/edge touch. In the moment when the upper wedge and the lower wedge touch in their bold-style edges, an intermediate vertex v of degree 8 is created. This vertex splits into two vertices of degree 3 when the upper wedge breaks into two pieces.

The three events in Figure 5 are unique. In conclusion, apart from the initial events there is only one surface event which leaves two choices – the saddle point vertex touch in Figure 4 (middle). This can also be inferred from the uniqueness of the corresponding valid bisector graphs. Observe that the bisector graphs for the degree-6 vertex touch and the edge/edge touch are disconnected. They are forests consisting of two trees each.

The edges and vertices of the shrinking polytope, by definition, trace out the facets and edges of the 3-dimensional straight skeleton. Therefore each event, either solid or surface event, constructs a new skeleton vertex v, which at that very moment is also a vertex of the shrinking polytope. Moreover, the facial structure of the local offset surface at v *after* the event directly specifies the way how to continue the construction of the straight skeleton locally at v. Recall that the skeleton facets and edges are portions of the respective angular bisector planes and trisector lines.

5. EXAMPLES

Our method for computing 3-dimensional straight skeletons has been implemented, and we applied it to various sample polytopes. Emphasis has been put on the correct implementation of the various events, and on the generation of valid local offset surfaces via bisector graphs.

Events can be detected either by predicting the occurrence of edge lengths zero on the polytope surface, or by predicting the collision of polytope faces. While the former task is local and efficient, the latter task basically requires a look at the whole component (a facet, or the entire polytope). We did not try to optimize the runtime, as we are for now mainly interested in the structural properties of the straight skeleton, and on its behavior under different valid choices. The current implementation needs (roughly) $O(e + r^2)$ time to find the next event, where e and r denote the number of edges (respectively, reflex edges) of the polytope at the present shrinking state.

We give a brief excerpt of the output, including 19 polytopes quite distinct in size and shape; see Table 1. When exploiting the different

Poly-ID	Name	Vertices	Edges	Facets
1	Cube shaken	8	12	6
2	Schönhardt	6	12	8
3	Iron Maiden	20	30	15
4	Verworrtakelt I	66	192	128
5	Verworrtakelt II	66	192	128
6	Star	110	324	216
7	Armadillo small	50	144	96
8	Armadillo	99	291	194
9	Asteroid	20	54	36
10	Stanford Bunny	152	450	300
11	Pawn (Chess)	42	119	79
12	Chinese Lion	89	261	174
13	Convex Piece	38	108	72
14	Hand	52	150	100
15	Iron Maiden II	32	90	60
16	Sphere shaken	66	192	128
17	Venus small	63	183	122
18	Venus	142	420	280
19	Sea Star	12	30	20

TABLE 1. Examples of input polytopes

choices for offset surfaces provided by all the valid bisector graphs, offsets (and skeletons) with tailor-made features can be generated. In particular, convex and reflex surface edges will arise in different ways. This might be desirable in a particular application. We tested the two extreme cases – minimizing and maximizing the number of occurring reflex edges. The first case tends to keep the polytope volume large, whereas in the other case the polytope is usually 'slimmed down' more quickly. Different straight skeletons for the same polytope are obtained, for which we counted the number of *nodes* (skeleton vertices, including the ones of the initial polytope), *arcs* (skeleton edges), and *sheets* (interior facets of the skeleton). Note that the number of nodes equals the number of initial vertices plus the number of events.

Consult Table 2. What can be observed? For most polytopes, maximizing the number of convex edges leads to fewer events, and thus to a combinatorially smaller straight skeleton. Polytope ID 13 is convex, so no difference occurs because the skeleton is the medial axis, a unique structure. Equality also holds for IDs 1, 3, and 9, because these nonconvex polytopes are small and do not leave choices. An exception is ID 15, where the number of events (and nodes) gets by one larger when convex edges are maximized.

ID	Nodes	Arcs	Sheets	Time [s]	Nodes	Arcs	Sheets	Time [s]
1	12	12	13	0.01	12	12	13	0.01
2	12	18	19	0.01	14	22	23	0.02
3	69	108	70	0.53	69	108	70	0.53
4	402	798	596	25.04	475	944	671	41.98
5	388	770	583	26.35	467	928	667	42.88
6	688	1370	1026	50.11	728	1450	1052	87.82
7	229	452	368	5.75	270	534	417	8.372
8	512	1018	802	28.42	587	1168	885	43.55
9	86	166	135	0.65	86	166	135	0.65
10	949	1892	1400	112.70	980	1954	1440	140.10
11	210	413	325	5.09	232	457	351	6.29
12	483	960	744	29.58	519	1032	781	41.10
13	196	386	299	3.13	196	386	299	3.13
14	238	470	386	7.25	287	568	439	10.62
15	324	642	412	35.18	323	640	411	33.97
16	335	664	522	9.02	341	676	529	9.39
17	344	682	526	12.89	411	816	597	20.88
18	948	1890	1376	139.10	1093	2180	1527	201.40
19	41	76	66	1.00	43	80	71	1.03

TABLE 2. Count of straight skeleton components when convex edges are maximized (left) or minimized (right).

Figure 6 displays a polytope (ID 19) with two saddle points of high degree. Although the polytope has only 12 vertices, the difference between the two extremal methods of shrinking can be seen.

In the presented examples, and also for most of our other test polytopes, the number of events needed is a small multiple (< 10) of the number n of polytope vertices. The known best theoretical upper bound is the trivial one only, $\binom{n}{4} = O(n^4)$. It stems from the fact that each event requires four offset planes to meet at the same point, which happens at most once during the entire shrinking process. This bound is not considered to be asymptotically tight. Slightly super-quadratic lower bound examples can be constructed however, see [7], which indicates that the straight skeleton – in the worst case – is possibly more complex than the medial axis, for which only examples of size $\Theta(n^2)$ are known.

There can be super-exponentially many (in n) different straight skeletons for a given polytope with n vertices. Finding skeletons efficiently that require a *minimum number of constructing events* (or even characterizing them in terms of bisector graphs) is an interesting and practically relevant question, as such skeletons have a smallest number of vertices. However, the polytope ID 15 disproves the (naive) conjecture that maximizing the number of convex edges leads to a minimum number of events.
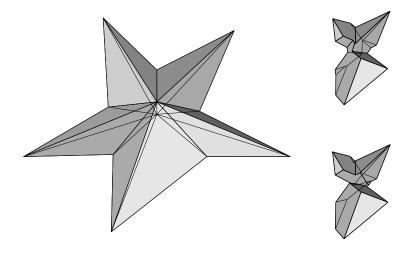
FIGURE 6. Sea star shrinking in two different ways.

REFERENCES

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

[2] W. Aigner, F. Aurenhammer, and B. Jüttler. On triangulation axes of polygons. *Proc. 28th European Workshop on Computational Geometry*, 2012, 25–128.

[3] O. Aichholzer and F. Aurenhammer. Straight skeletons for general polygonal figures in the plane. Proc. 2nd Ann. International Computing and Combinatorics Conference, Springer Lecture Notes in Computer Science 1090, 1996, 117–126.

[4] D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes—a state-of-the-art report. In: Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration, T. Müller, B. Hamann, B. Russell (eds.), Springer Series on Mathematics and Visualization, 2008, 109–125.

[5] F. Aurenhammer, R. Klein, and D.T. Lee. *Voronoi Diagrams and Delaunay Triangulations.* World Scientific, Singapore, 2013.

[6] F. Aurenhammer and G. Walzl. Structure and computation of straight skeletons in 3-space. Proc. 24th International Symposium on Algorithms and Computation, 2013; to appear.

[7] G. Barequet, D. Eppstein, M.T. Goodrich, and A. Vaxman. Straight skeletons of three-dimensional polyhedra. Proc. 16th Ann. European Symposium on Algorithms, Springer Lecture Notes in Computer Science 5193, 2008, 148–160.

[8] D. Eppstein and J. Erickson, Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. Discrete & Computational Geometry 22 (1999), 569-592.

[9] J. Martinez, M. Vigo, and N. Pla-Garcia. Skeleton computation of orthogonal polyhedra. Computer Graphics Forum 30 (2011), 1573–1582.

[10] F. Ruskey and A. Williams. Generating balanced parentheses and binary trees by prefix shifts. Proc. 14th Australasian Theory Symposium, 2008, 22-25.

[11] K. Siddiqi and S.M. Pizer. *Medial Representations. Mathematics, Algorithms, and Applications.* Springer Series on Computational Imaging and Vision 37, 2008.

[12] A. Vigneron and L. Yan. A faster algorithm for computing motorcycle graphs. Proc. 29th Ann. ACM Symposium on Computational Geometry, 2013, 17–26.

INSTITUTE FOR THEORETICAL COMPUTER SCIENCE, GRAZ UNIVERSITY OF TECHNOLOGY, INFFELDGASSE 16B, 8010 GRAZ, AUSTRIA

E-mail address: `auren@igi.tugraz.at`

INSTITUTE FOR THEORETICAL COMPUTER SCIENCE, GRAZ UNIVERSITY OF TECHNOLOGY, INFFELDGASSE 16B, 8010 GRAZ, AUSTRIA

E-mail address: `gernot.walzl@igi.tugraz.at`