Generalizing Monotonicity: On Recognizing Special Classes of Polygons and Polyhedra

Prosenjit Bose

Marc van Kreveld

institute of information and computing sciences, utrecht university

technical report UU-CS-2005-042

www.cs.uu.nl

Generalizing Monotonicity: On Recognizing Special Classes of Polygons and Polyhedra

Prosenjit Bose
School of Computer Science
Carleton University
jit@scs.carleton.ca

Marc van Kreveld
Inst. of Information and Computing Sciences
Utrecht University
marc@cs.uu.nl

We dedicate this paper to Godfried Toussaint on the occasion of his 60th birthday.

Abstract

A simple polyhedron is *weakly-monotonic* in direction $\vec{d}$ provided that the intersection of the polyhedron and any plane with normal $\vec{d}$ is simply-connected (i.e. empty, a point, a line-segment or a simple polygon). Furthermore, if the intersection is a convex set, then the polyhedron is said to be *weakly-monotonic in the convex sense*. Toussaint [10] introduced these types of polyhedra as generalizations of the 2-dimensional notion of monotonicity. We study the following recognition problems:

Given a simple n-vertex polyhedron in 3-dimensions, we present an $O(n \log n)$ time algorithm to determine if there exists a direction $\vec{d}$ such that when sweeping over the polyhedron with a plane in direction $\vec{d}$, the cross-section (or intersection) is a convex set. If we allow multiple convex polygons in the cross-section as opposed to a single convex polygon, then we provide a linear-time recognition algorithm. For simply-connected cross-sections (i.e. the cross-section is empty, a point, a line-segment or a simple polygon), we derive an $O(n^2)$ time recognition algorithm to determine if a direction $\vec{d}$ exists.

We then study variations of monotonicity in 2-dimensions, i.e. for simple polygons. Given a simple n-vertex polygon P, we can determine whether or not a line ℓ can be swept over P in a continuous manner but with varying direction, such that any position of ℓ intersects P in at most two edges. We study two variants of the problem: one where the line is allowed to sweep over a portion of the polygon multiple times and one where it can sweep any portion of the polygon only once. Although the latter problem is slightly more complicated than the former since the line movements are restricted, our solutions to both problems run in $O(n^2)$ time.

1 Introduction

Determining if a given polygon or polyhedron possesses a certain property such as convexity or monotonicity has been a well-studied problem in the literature from both a theoretical and practical point of view (see [11, 12] for several surveys and application areas). Essentially, when an object possesses a particular geometric property, often this property can be exploited to simplify as well as speed-up algorithms dealing with such objects. These techniques have been used in many different areas such as graphics, manufacturing, and geographic information systems to name a few. For example, determining the longest line segment inside a convex polyhedron is a much simpler problem to solve than determining the longest line segment inside an arbitrary polyhedron. Triangulating a monotone polygon is much easier than triangulating an arbitrary polygon.

In this paper, we address problems dealing with recognizing whether or not polygons and polyhedra have a certain property related to monotonicity. Recall that a simple polygon P is monotone in direction $\vec{d}$ if the intersection of P with any line ℓ having normal $\vec{d}$ is a convex set (i.e. empty, a point, or a line-segment). Alternatively, this can be viewed as sweeping a line ℓ over P in direction $\vec{d}$ such that the intersection is always a convex set. Since both views of this problem are equivalent, we use them interchangeably throughout.

Preparata and Supowit [8] presented a linear time algorithm to report all directions in which a given polygon is monotone. Rappaport and Rosenbloom [9] subsequently generalized this to report all the ways of decomposing a polygon into two monotone chains that are not necessarily monotone in the same direction. Toussaint [10] studied a three-dimensional generalization of the above definition of monotonicity. He defined a polyhedron P to be *weakly-monotonic* if there exists a direction $\vec{d}$ such that the intersection of P with any plane H with normal $\vec{d}$ is simply-connected (i.e. convex set, or a simple polygon). In 3-dimensions, the alternate view is the existence of a direction $\vec{d}$ such that a plane H with normal $\vec{d}$ can be swept over P in direction $\vec{d}$ with the property that the intersection of H with P is simply-connected throughout the sweep. Since there are many different types of polygons, Toussaint went on to include the type of polygon in the definition of monotonicity. So, if all polygons in the intersections are convex, then the polyhedron is *weakly-monotonic in the convex sense*. Similarly, if the polygons are all simple, then the polyhedron is *weakly-monotonic in the simple sense* and so on. One of the open problems Toussaint posed was to efficiently recognize polyhedra that are weakly-monotonic in the convex sense and the simple sense. We resolve both these recognition problems.

The first question we address is recognizing if a polyhedron is weakly-monotonic in the convex sense. Given an n-vertex polyhedron, we provide an $O(n \log n)$ time recognition algorithm. Specifically, we give an $O(n \log n)$ time algorithm to find a vector $\vec{d}$ if one exists, such that the intersection between a simple n-vertex polyhedron and any plane with normal $\vec{d}$ is a convex set. Our algorithm reports all directions from which a polyhedron is weakly-monotonic in the convex sense. If we allow the cross-section to contain the disjoint union of several convex sets as opposed to a single convex set, then we can solve the problem in linear time.

The second question we address is the recognition of polyhedra that are weakly-monotonic in the simple sense, i.e. there exists a direction $\vec{d}$ such that the intersection of the polyhedron and any plane with normal $\vec{d}$ is a convex set or a simple polygon. We provide an $O(n^2)$ time algorithm to find such a direction if one exists. Note that the cross-section may not become disconnected, nor may it contain a holes, it must be simply-connected.

Finally, we consider 2-dimensional generalizations of monotonicity. Given a simple n-vertex polygon P, we wish to determine if a line can be swept over the whole polygon P such that every cross-section is a convex set (i.e. empty, a point or a line-segment). If a polygon possesses this property, we call it *sweepable*. Now, if P is monotone, then clearly it is sweepable and the direction $\vec{d}$ of monotonicity provides the direction in which the line can be swept over the polygon. We are generalizing the notion of monotonicity by allowing the line to change orientation as it is being swept over the polygon. Therefore, there can be simple polygons that are not monotonic in any direction but are still *sweepable*. Note that there are two natural classes of sweepable polygons. One where the polygon is sweepable but certain portions of the polygon must be swept over multiple times. The other is when the line is swept over the polygon and no portion of the polygon is swept over more than once. If a polygon is sweepable in the latter way, we call it *strictly-sweepable*. We present $O(n^2)$ time algorithms to determine if a simple n-vertex polygon is sweepable and strictly-sweepable. The latter problem is slightly more complicated than the former since the line movements are restricted because we cannot go over a portion of the polygon that was already swept. Our algorithms output all the movements a line can make to complete the sweep both in the sweepable setting and the strictly-sweepable setting.

2 Recognizing Weakly-Monotonic Polyhedra in the Convex Sense

Given a simple polyhedron[1] P with n vertices in 3-space, we want to determine if it is weakly-monotonic in the convex sense. In order to recognize such polyhedra, we need to determine if there is a sweep direction $\vec{d}$ such that a plane H with normal $\vec{d}$ can be swept over P in direction $\vec{d}$ with the property that every cross-section (i.e. intersection of P with H) is simply-connected (i.e. convex set or a convex polygon). A slight variation to this question is to allow every cross-section to contain multiple convex polygons as opposed to a single convex polygon. It turns out that this variation is simpler to solve. We call such a polyhedron *sweepable* in direction $\vec{d}$. We first show how to determine if a polyhedron is sweepable in a given direction. We then show how to determine if it is weakly-monotonic in the convex sense. We begin by establishing that only the reflex edges of the polyhedron need to be considered in order to solve both these problems.

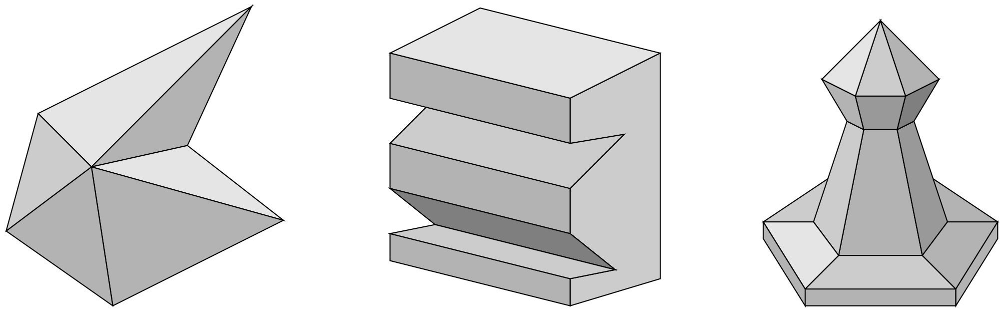

Figure 1: Polyhedra with convex cross-sections only.

Lemma 1 *Let P be a simple polyhedron and let E_r be the set of its reflex edges. P is sweepable in direction $\vec{d}$ such that every cross-section is the disjoint union of one or more convex sets if and only if a sweep plane with normal $\vec{d}$ is parallel to all edges in E_r.*

Proof: If a plane intersects a reflex edge of P properly, then that cross-section contains a reflex vertex at the proper intersection. If a cross-section has a reflex vertex, then it is either caused by a proper intersection with a reflex edge, or by a vertex of P. In the former case the lemma is obviously true, so assume a vertex v of P lies in the cross-section. Since P is not flat, it extends to the one or the other side of the plane that created the cross-section. Move the plane in that direction a small amount, not so far that it would pass over or contain another vertex of P. Then the plane necessarily intersects a reflex edge of P incident to v. $\triangle$

Note that if a polyhedron is *sweepable* in direction $\vec{d}$, then it is not necessarily weakly-monotonic in direction $\vec{d}$ since there can be cross-sections that consist of more than one convex polygon. However, these cross-sections will never contain a reflex vertex. Multiple convex polygons can occur in the cross-section when the sweep plane contains a reflex edge, and both facets incident to the reflex edge are to the same side of the sweep plane. Figure 1 shows some polyhedra that are sweepable, where certain sweep directions only give single convex polygons as the cross-section, whereas other sweep directions may give several convex polygons in some cross-section. Note that all essential changes to the cross-section occur when the sweep plane reaches or passes a vertex.

To decide algorithmically if a polyhedron is sweepable in some direction, we distinguish four cases:

[1]See the chapter by E. Schulte in O'Rourke and Goodman[11] for a detailed definition of simple polyhedron

1. Polyhedron P is convex.

2. Polyhedron P has at least one reflex edge, and all reflex edges have the same orientation (i.e. they are all parallel to some line).

3. Polyhedron P has at least two reflex edges, and all reflex edges of P have orientations that span a plane.

4. Polyhedron P has three reflex edges that are linearly independent.

It is simple to determine in linear time which of the four cases applies for a given polyhedron. Cases 1 and 4 can be handled trivially: In case 1 every sweep direction works, and in case 4 no sweep direction works, which follows from Lemma 1. Note that these two cases are equivalent whether we are trying to determine if a polyhedron is sweepable or we are trying to determine if a polyhedron is weakly-monotonic in the convex sense.

Cases 2 and 3 are trivial if multiple convex polygons are allowed in the cross-section. For both cases, by the lemma, we conclude that the polyhedron is sweepable. Therefore, to determine if a polyhedron is sweepable, we need to establish which of the 4 cases applies. This decision can be made in linear time.

We now address the problem of determining if a polyhedron is weakly-monotonic in the convex sense. In this setting, the intersection must always be at most one convex polygon, which is the only additional condition that must be tested. Therefore, all directions where the sweep plane contains a reflex edge and both incident faces are to the same side, are forbidden. We noted above that we only need to concentrate on case 2 and 3. If case 3 applies, there is only one candidate orientation for a sweep plane, the orientation that contains all reflex edges of P. Now, we can test this orientation with the extra condition that the cross-section can only contain a single convex polygon easily in linear time.

If case 2 applies, the sweep direction must be normal to the reflex edge(s) of P, which implies that the sweep plane still has one rotational degree of freedom. This can be represented by a circle of candidate sweep directions. Each reflex edge eliminates two antipodal intervals (circular arc) from the circle of orientations to fulfill the condition that both incident faces may not be on the same side of the sweep plane when the sweep plane contains that edge. To test if all circular arcs together cover the circle, implying that no good sweep direction exists, we solve the problem by sorting the endpoints of the circular arcs along the circle in $O(n \log n)$ time.

Theorem 1 *Let P be a simple polyhedron with n vertices. In linear time, one can test if a sweep direction exists so that every cross-section is a collection of convex polygons. In $O(n \log n)$ time, one can test if a sweep direction exists so that every cross-section is a single convex polygon.*

Corollary 1 *In $O(n \log n)$ time, we can decide if an n-vertex polyhedron is weakly-monotonic in the convex sense.*

3 Recognizing Weakly-Monotonic Polyhedra in the Simple Sense

In this section we wish to determine whether a simple polyhedron P admits a sweep direction so that every cross-section is simply-connected (i.e. convex set or a simple polygon). To this end, we outline what topological changes can occur when sweeping a simple polyhedron by a plane in a specified direction. Assume we sweep vertically, from top to bottom, with a horizontal plane. The possible changes when sweeping in direction $-\vec{z}$ are (see Figure 2):

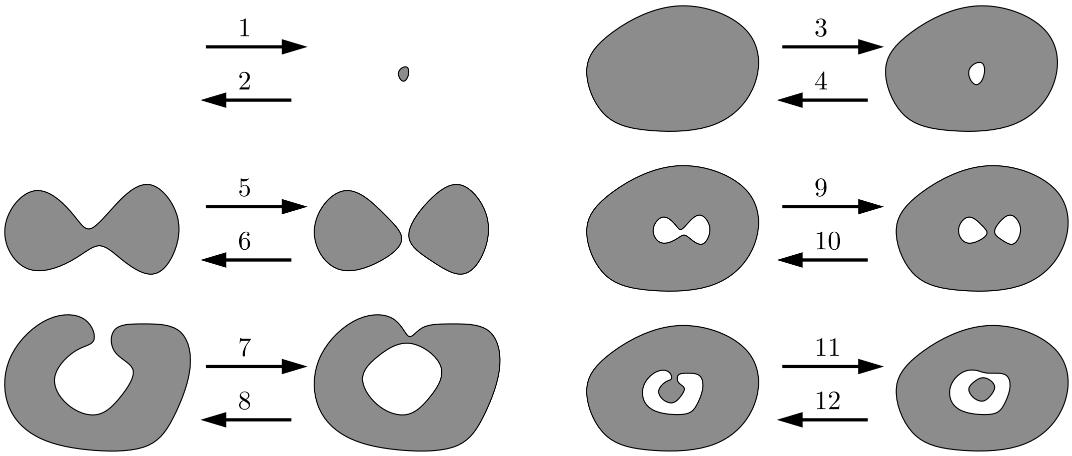

Figure 2: Topological changes that can occur in the cross-sections.

1. A new component in the cross-section starts. This is caused by a local maximum of P in direction $\vec{z}$.
2. A component in the cross-section disappears. This is caused by a local minimum of P in direction $\vec{z}$.
3. A component in the cross-section creates a hole 'in the middle'. This is caused by a local maximum of the complement of P.
4. A component in the cross-section closes a hole 'in the middle'. This is caused by a local minimum of the complement of P.
5. A component splits into two components. This is caused by a saddle-type vertex.
6. Two components touch to form one component. This is caused by a saddle-type vertex.
7. A component in the cross-section creates a hole 'at the boundary', that is, the outer boundary of a component curls to touch itself, thereby creating a hole. This is caused by a saddle-type vertex.
8. A component in the cross-section opens a hole 'at the boundary'.
9. A hole of a component splits into two.
10. Two holes touch to form one hole.
11. A hole has a new component split off inside the hole.
12. A component inside a hole merges with a surrounding component.

The fact that these are the only topological changes that can occur follows from Morse Theory (see Milnor [7] for details). All of these changes can only occur when the sweep plane reaches or passes a vertex. Assume for a moment that the sweep plane is not parallel to any edge. For every vertex, the local situation determines what type of topological change occurs, if any. With 'local situation' we mean that it only depends on the adjacent facets and edges of the vertex, and nothing else. If there is one edge e parallel to the sweep plane, the topological change that occurs, if any, when the sweep plane passes e depends on all facets and edges incident to both endpoints of the edge e.

We represent all possible sweep directions by a sphere centered at the origin o. A point q on the sphere represents the sweep direction $\vec{oq}$. Each vertex of polyhedron P defines a partition of the sphere of directions into regions where some topological change occurs in the cross-section at that vertex. A vertex can be a local maximum in a certain sweep direction; this concept was studied and used for the application of mold filling in Bose et al. [1]. Note that the same vertex can be one where in another sweep direction a hole is created or destroyed in the cross-section and in another direction may cause no topological change at all. See Figures 3 and 4.

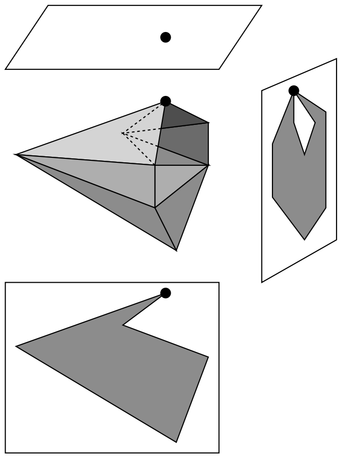

Figure 3: A vertex that causes different types of topological change for different sweep directions

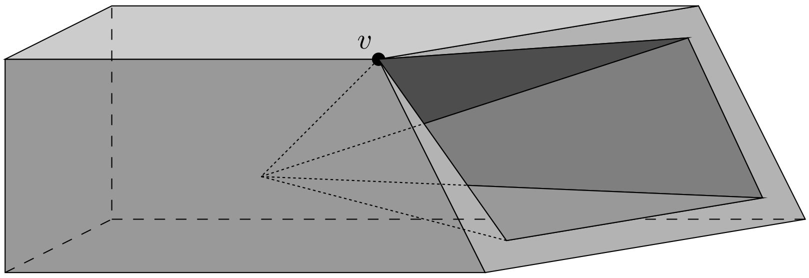

Figure 4: A close-up view of the vertex in Figure 3.

On the sphere of directions, the boundaries between different regions defined by a vertex are determined by the directions of the edges incident to the vertex. Every edge in fact gives rise to a great circle on the sphere of directions which represents the collection of normals of the sweep plane for which it is parallel to the edge. In other words, the circle represents the perpendiculars to the edge.

Since a polyhedron with n vertices has $O(n)$ edges, the sphere of directions is partitioned into $O(n^2)$ regions bounded by pieces of great circles. Each region defines a set of directions for which a sweep encounters the same topological changes. We can compute this arrangement by standard methods in quadratic time. The process is outlined in detail in Bose et al. [1].

Notice that a sweep in a direction $\vec{d}$ has at most one single simple polygon in the cross-section if and only if it has one local maximum and one local minimum in that direction, and no other topological changes occur. The number of local maxima and minima, and the occurrence of topological changes are interdependent. A polyhedron that has one local maximum and two local minima must have a vertex at which the cross-section is split, one part of the cross-section ending at the one local minimum and the other ending at the other. This characterization is at the heart of our algorithm. We prove it in the lemma below.

Lemma 2 *Given a simple polyhedron P and a sweep direction $\vec{d}$, every cross-section with a plane normal to $\vec{d}$ is convex set or a simple polygon if and only if P has exactly one local maximum and one local minimum in direction $\vec{d}$, and no local maximum or minimum of the complement of P in direction $\vec{d}$.*

Proof: The 'only if' is easy. For the 'if'-part, observe the following when sweeping P in a given direction. A topological change that makes the cross-section disconnected (nos. 1 and 5 in Figure 2) is either caused by a second local maximum, or it implies two local minima (if those pieces stay apart), or that P has higher genus (if the piece merge later). All three are ruled out by the assumptions that P is simple and has exactly one local maximum and one local minimum.

The only other topological change that can occur is the appearance of a hole inside the polygon. Such a hole can be created in the middle (no. 3 in Figure 2; by a local maximum of the complement of P) or at the boundary (no. 7). The former case is ruled out by assumption in the lemma statement, so assume the latter occurs. This hole must eventually disappear from the cross-section, which on its turn can occur in the middle (no. 4; by a local minimum of the complement of P) or at the boundary (no. 8). Again the former case is ruled out. But if a hole is created at the boundary and disappears at the boundary, then P must have higher genus and hence, cannot be simple. $\triangle$

To determine a sweep direction in which every cross-section is simply-connected, we only have to consider local maxima and minima of the interior and the exterior of P. Every vertex determines a (possibly empty) region on the sphere of directions such that it is a local maximum (or minimum) for those sweep directions. In the same way as in Bose et al. [1], we can determine this in quadratic time. We determine a cell in the arrangement on the sphere of directions that corresponds to the smallest number of local maxima plus minima (interior and exterior). If there is one local maximum and one local minimum of the interior, this smallest number is 2, and only then is the polyhedron sweepable with simply-connected cross-sections.

Theorem 2 *Let P be a simple polyhedron with n vertices. In quadratic time, one can find all the sweep directions having the property that every cross-section is simply connected or empty.*

Corollary 2 *In $O(n^2)$ time we can determine if an n-vertex simple polyhedron is weakly-monotonic in the simple sense. We can output all directions from which the polyhedron is weakly-monotonic.*

4 Recognizing Sweepable and Strictly-Sweepable Polygons

In this section, we consider 2-dimensional generalizations of monotonicity. Given a simple n-vertex polygon P, we want to determine if a line can be swept over the whole polygon P such that every cross-section is convex set (i.e. empty, a point or a line segment). This is a generalization of the standard notion of a monotone polygon because we allow the line to change orientation as it is being swept over the polygon. We model the movements of the sweep line by a sequence of translations and rotations that alternate. For rotations we specify a center of rotation on the sweep line, and an angle of rotation. We start out with a simple polygon P in the plane, and a directed horizontal line strictly above all vertices of the polygon. The final situation of the sweep is one where the directed line is strictly below all vertices of the polygon. The line must be directed, otherwise it could go from initial to final position using a half turn beside the polygon, not intersecting it at all.

The simple polygon P is assumed to be a closed set with interior and boundary. During the translations and rotations, the sweep line may only intersect P in a single line-segment, or a point, or not at all. So when the sweep line intersects the boundary of P it generally does so at two points. Figure 5 shows that this kind of sweep is more general than sweeping by translation only (i.e. monotone polygon). As noted in the introduction, we study two versions of the problem.

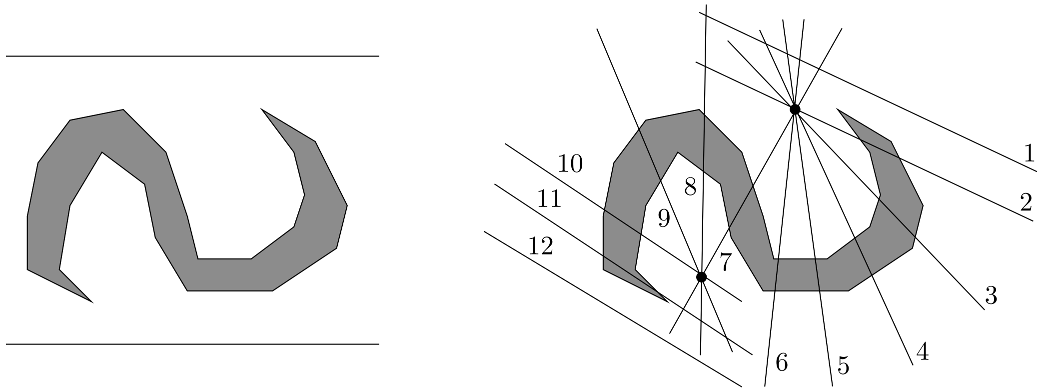

Figure 5: The top line shows the position before the sweep; the bottom line after the sweep. Several possible intermediate positions are shown in the figure on the right.

One where the sweep line is not allowed to sweep over any portion of the interior of the polygon more than once and one where the line may sweep over portions of the polygon multiple times. Recall that in the latter case, the polygon is said to be sweepable and in the former case, it is strictly-sweepable. We develop an algorithm to decide if a given polygon is strictly-sweepable or sweepable. Our algorithm outlines the sequence of translations and rotations when the polygon is sweepable or strictly-sweepable. Note that this is related to but different from *walkable polygons* [5] and *street polygons* [6].

Simple polygons that are sweepable are necessarily *street polygons* [6]. However, not all street polygons are sweepable, for example, a simple polygon that is spiraling. Furthermore, sweepable polygons are *walkable polygons* [5], but the reverse is not necessarily so, again because of spiraling polygons. The difference between *general walkable* and *straight walkable* is very similar to the difference between sweepable and strictly-sweepable.

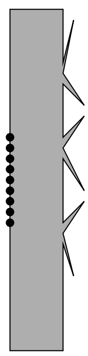

Figure 6: The sweep line must go over vertices $\Omega(n^2)$ times.

The complexity of a sweep can be expressed in the number of elementary motions of the sweep line. One can also look at the number of times the sweep passes a vertex. Since the sweep may only be possible if the sweep line 'goes back' over vertices, we potentially have a large number of crossings between the sweep-line and a vertex. Figure 6 shows that the sweep line may be forced to sweep over a linear number of different vertices a linear number of times.

The key to the solution is dualization. We use the standard duality transform that maps a point $p = (a, b)$ to a line $D_p = \{(x, y) | y = ax - b\}$ and the line $L = \{(x, y) | y = mx + c\}$ to the point $(m, -c)$. This duality transformation preserves incidences and above/below relations (see deBerg et al. [2] for details). In the dual arrangement of a set of points, we call the 'topmost' face, the unbounded face defined by the upper envelope of the dual lines and the 'bottommost' face, the unbounded face of the arrangement defined by the lower envelope of the dual lines. We require that the sweep line always intersect the polygon in at most two edges, except when going over vertices. In the dual, the edges of the polygon are double wedges, and the sweep line is a point. Translations of the sweep line equate to the dual 'sweep point' translating vertically. Rotations of the sweep line equate to the sweep point translating along a non-vertical line. When the sweep line goes from one edge to the next over a vertex, the dual sweep point goes from one double wedge to another via their common bounding line.

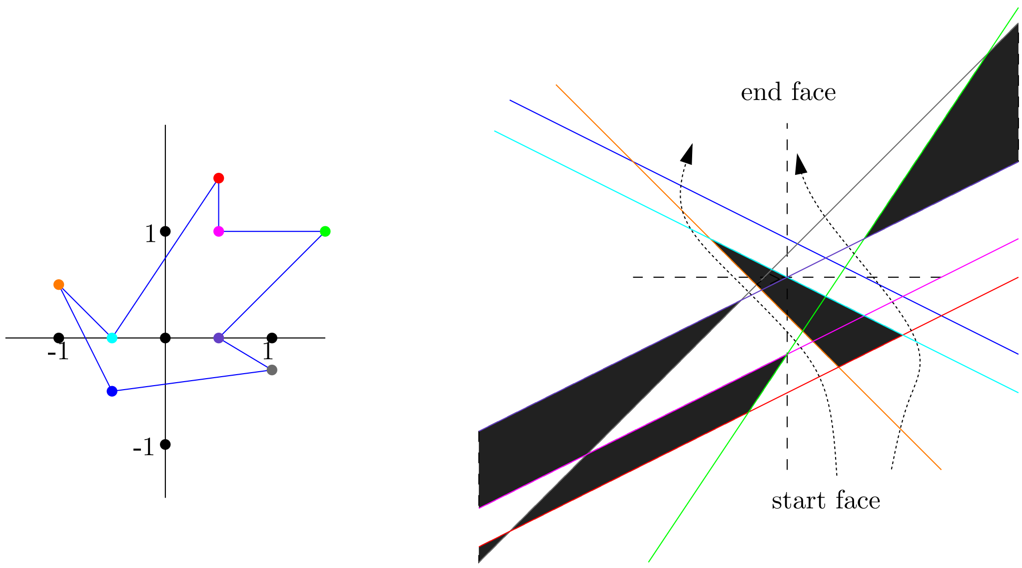

Figure 7: A simple polygon and the dual arrangement of its vertices. Forbidden faces are gray; two valid paths—through white faces only—are shown

The vertices of the polygon dualize to all relevant lines in the dual plane. The bottommost face is the one where the sweep point starts as it is the placement of a line in the primal with all vertices of the polygon above the line. The topmost face is where the sweep point must end since it represents a placement of the sweep line in the primal with all the polygon vertices lying below the sweep line. If the dual sweep point lies inside more than two double wedges, then there are more than two edges of the polygon that intersect the sweep line. So the question is whether the sweep point can go from the bottommost face to the topmost face via faces that lie in at most two double wedges. From one face, the sweep point can only go to adjacent faces that share an edge on their boundary. It cannot go from one face to another that only share a vertex.

As noted in the beginning of this section, if we are given a line with all polygon vertices lying above it, one can easily move it continuously to a position where all polygon vertices lie below it without ever intersecting the polygon. To force the sweep line to sweep over the polygon, we need to consider oriented lines. Given an oriented line, where the polygon vertices lie to the right, the only way to continuously move the oriented line where all polygon vertices lie to the left is by sweeping over the polygon.

However, two lines at the same position but opposite orientation should not dualize to the same point. To handle oriented lines, we use two copies of the dual plane, one for leftward directed lines and one for rightward directed lines, see Figure 8. The two arrangements are called $\mathcal{A}_1$ and $\mathcal{A}_2$;

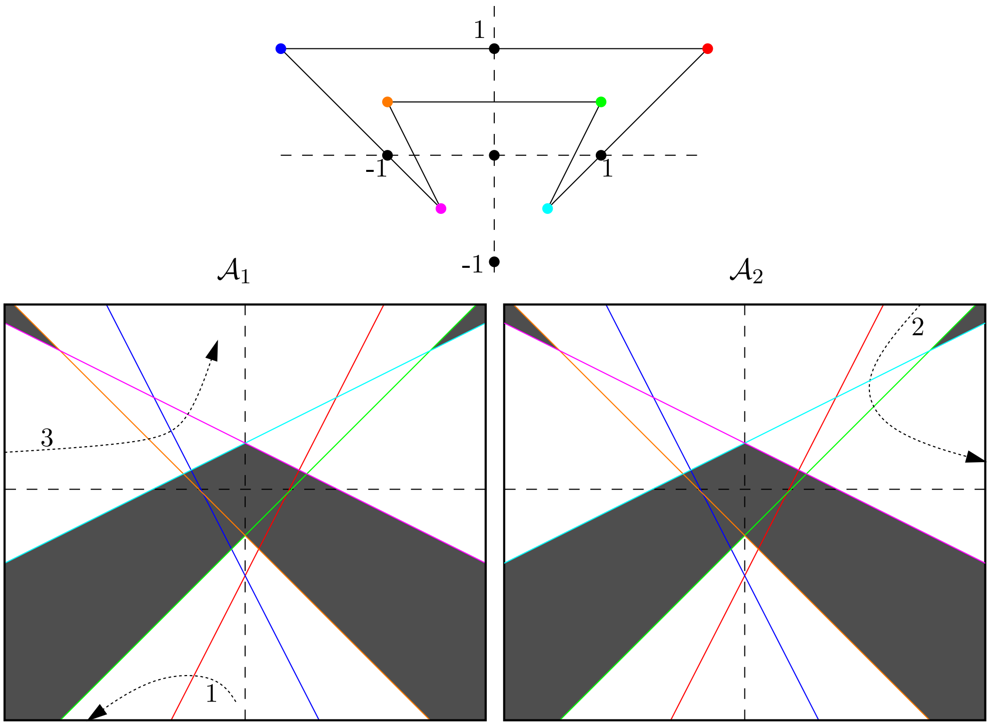

Figure 8: A simple polygon and the dual arrangements $\mathcal{A}_1$ and $\mathcal{A}_2$. Any valid sweep over the polygon (for example, the one shown by the arrows 1, 2, 3) must go via faces of both arrangements.

they are identical. Within one arrangement, the point dual to the sweep line can go from one face to an adjacent face over a shared edge. Each unbounded face in $\mathcal{A}_1$ is connected to the unbounded face in $\mathcal{A}_2$ that is on the opposite side of the arrangement. The point dual to the sweep line can move between these faces, which corresponds to the situation where the sweep line rotates past the vertical direction. The goal now is to find a valid path for a point in the bottommost face in $\mathcal{A}_1$ to the topmost face in $\mathcal{A}_1$, but the point may move via $\mathcal{A}_2$.

At the start of the algorithm, we first rotate P slightly so that no two vertices have the same x-coordinate. This makes sure that we do not have parallel lines in the dual arrangements to be computed. Each of the two dual arrangements can be computed with standard techniques in quadratic time, and we can also predetermine with every face in how many double wedges it lies (Chapter 8 in deBerg et al. [2]). Faces in more than two double wedges are forbidden. Then the question whether P is sweepable becomes that of going from the bottom face of $\mathcal{A}_1$ to the top face of $\mathcal{A}_1$ via a sequence of adjacent faces that are not forbidden. Answering this question in the dual arrangements is easy; it simply is a depth-first search on the faces. The induced graph $\mathcal{G}$ has $O(n^2)$ nodes and arcs and is planar. This solves the version of sweeping where the sweep line is allowed to go back over vertices in $O(n^2)$ time in total.

Theorem 3 *Given a simple polygon P with n vertices, in $O(n^2)$ time we can determine whether or not P is sweepable. That is, we can determine if a sweep with a line ℓ over P exists such that ℓ intersects P in at most one connected component.*

Now, we consider the version of the problem where the sweep line may not sweep over any point of P more than once. What does it mean in the dual that the sweep line is not allowed to sweep over a point twice? If a sweep line sweeps an edge in a forward direction, the dual sweep point

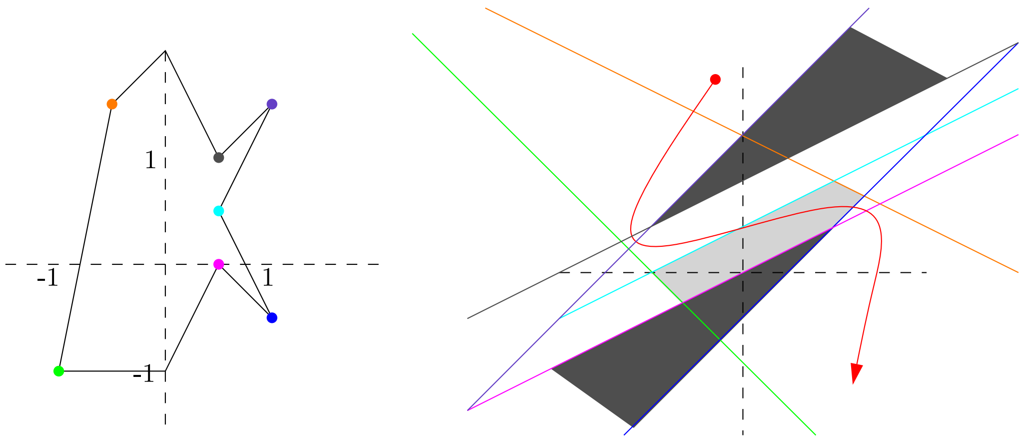

Figure 9: A simple polygon and a dual arrangement (present in duplicate). The light grey region (actually white) shows a face that cannot be traversed as shown if the sweep line may not go back.

moves in a restricted manner inside the double wedge dual to the edge. Consider an imaginary line through the sweep point and the apex of the double wedge. This line rotates about the apex when the sweep point moves. The forward sweep in primal plane dualizes to a translating point where the imaginary line only increases (or only decreases) its slope as long as the sweep point is inside the double wedge. Since the valid sweep lines we consider intersect at most two edges of the polygon at any time, the dual sweep point lies in only two double wedges. It is clear that if the sweep line does not go back on the edges of the polygon, it does not go back over any point of the interior either.

Lemma 3 *Let P be a polygon, e and e' two of its edges. The region in the dual plane representing all valid sweep lines intersecting e and e' either (i) is bounded and convex, or (ii) consists of two unbounded convex faces that are opposite in the arrangement.*

It is possible that the sweep line needs to go back over part of an edge, but without going back over a vertex of P again. For example, if the row of vertices on the left vertical edge of the polygon in Figure 6 were not present, then no forward sweep exists, but no vertex of P need be crossed backwards. Hence, the problem cannot be solved on the adjacency graph $\mathcal{G}$ of the dual arrangements $\mathcal{A}_1$ and $\mathcal{A}_2$ that we used before. We use a different, refined arrangement where going backwards implies crossing a line of the arrangement again. We take all extensions of edges at reflex vertices inside P, and place an extra vertex where the extension hits the boundary of P. We claim that in such a polygon, a forward sweeping motion exists if and only if every vertex is crossed exactly once. So we simply look for a path in the refined dual arrangements which crosses every line exactly once, which implies that the sweep line passes over every polygon vertex, including the internal projections, exactly once.

Lemma 4 *Let P be a simple polygon, and let P' be P extended with the vertices obtained when edges are extended at all reflex vertices. P' allows a sweep without going back over vertices if and only if P allows a sweep without going back over any point of P twice.*

Proof: One direction is trivial. If P allows a sweep without going back over any point of P twice, then P' allows the same since we simply added *dummy* vertices onto edges of P to build P'. We now prove the other direction.

When a sweep starts and proceeds while intersecting only two edges at the same time, we want to guarantee that the sweep line never has to trace back a little on either edge to be able to go

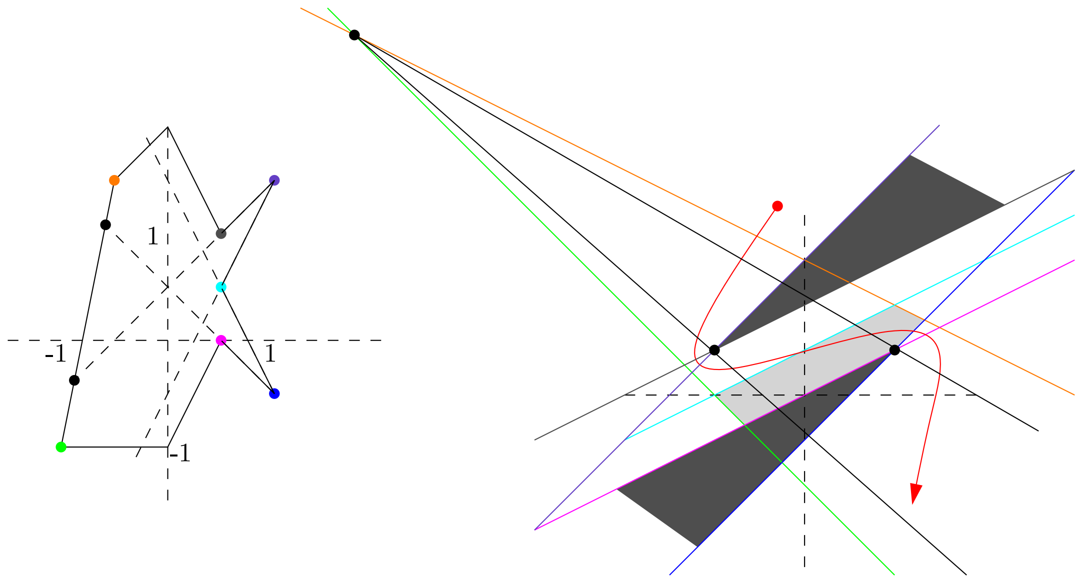

Figure 10: The example of Figure 9 with the extensions in primal and dual shown.

forward later. We must show that if the sweep line must go back, it will have to cross back over a vertex that was already passed.

Let e_i and e_j be the two edges of P that are intersected by the sweep line ℓ when it cannot proceed without first going back on e_i or e_j. The sweep cannot proceed when any forward motion makes that it intersects more edges of P, which implies that it passes over a vertex v that is not an endpoint of e_i or e_j. There are two cases: (1) v lies in between the intersection points of ℓ with e_i and with e_j. (2) v lies to one side of both intersection points. See Figure 11.

In case (1), note that v is a reflex vertex of P and that both incident edges lie on the unswept side of ℓ. Hence, v gave rise to two additional vertices from the extensions of these edges. We call them v' and v''. They both lie on the swept side of ℓ, and the only way that ℓ can eventually pass v is by going back all the way over v' or v''. So, the sweep line must go back over a vertex.

In case (2), v lies to one side of both e_i and e_j. Both edges incident to v must lie on the unswept side of ℓ. In this case the sweep is not stuck yet, because ℓ can pivot on v while going forward on both intersected edges. But ℓ is stuck if one of two additional situations arise, see the middle two pictures in Figure 11: (2a) There is a vertex w on the other side of e_i and e_j as v. (2b) There is a vertex u on the same side of e_i and e_j as v. In case (2a) the edges incident to w lie on the unswept side of ℓ. Now no sweep that simultaneously intersects e_i and e_j before meeting at least one of v and w can be good, so the sweep must go back over the upper endpoint of e_i or e_j (in fact, it must go back over all swept vertices). In case (2b), there is no good sweep for P at all. This follows because any line that contains v must intersect P in at least two connected components, which immediately implies that P is not sweepable.

There is a case (2c), also shown in Figure 11, where a vertex prevents the pivoting about v to be possible. In this case there is another vertex v' to the same side of e_i and e_j as v, but closer. In this case the pivoting can proceed about v' and vertex v is 'abandoned'.

In all cases where the sweep is stuck, either P is not sweepable, or the sweep line has to go back over at least some vertex to complete the sweep, which proves the lemma. △

The algorithm is as follows. We start by computing the extensions of all edges at reflex vertices by brute force. We do this rather than use a ray-shooting data structure since the overall running

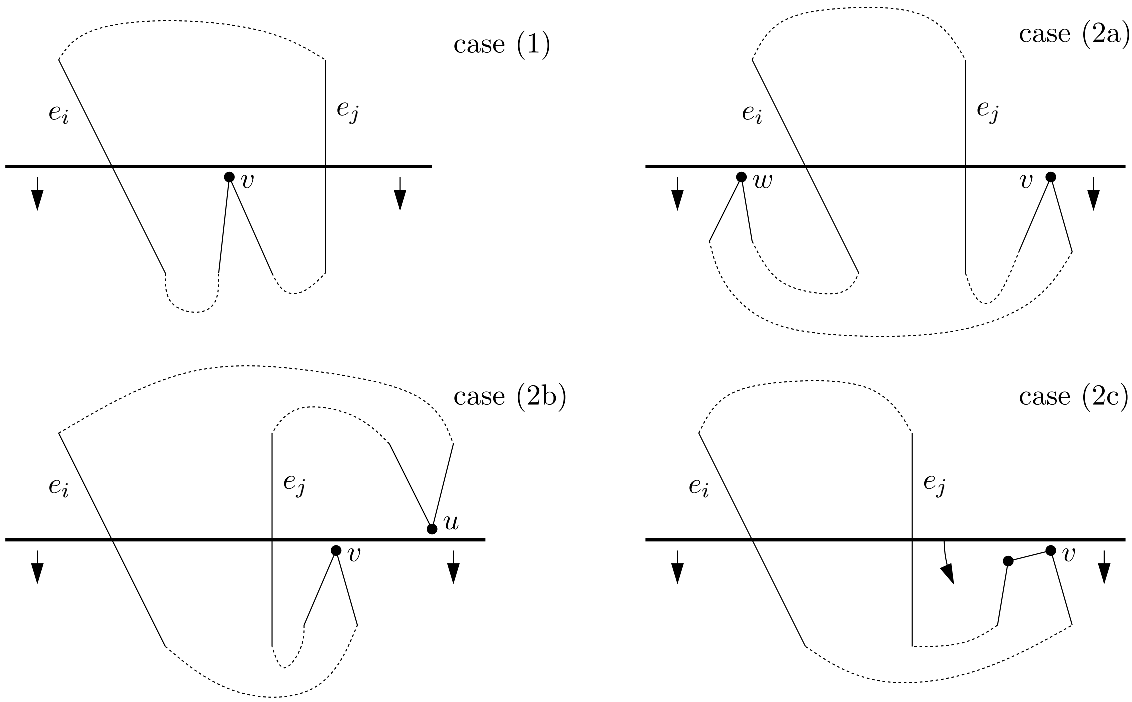

Figure 11: Illustration of the proof.

time of the algorithm is $O(n^2)$. This gives us the simple polygon P' with $O(n)$ extra vertices on the edges in $O(n^2)$ time; Let P' have m vertices. Then $m \leq 3n - 6$, because P has at least three convex vertices, and any reflex vertex gives two new vertices in P'.

Next, we rotate P' to make sure that no two vertices of P' have the same x-coordinate. We dualize the vertices of P', construct the arrangements $\mathcal{A}'_1$ and $\mathcal{A}'_2$ and the forbidden faces in these arrangements as before, and find a path from the lowest face to the highest face. Then we determine if there is a path from the lowest face in $\mathcal{A}'_1$ to the highest face in $\mathcal{A}'_1$ through adjacent faces in $\mathcal{A}'_1$ and $\mathcal{A}'_2$ that crosses exactly m lines. If such a path exists, then clearly the corresponding motion of the sweep line does not go back over vertices. Otherwise, no motion of the desired type exists. Finding a shortest path in a graph with $O(m^2)$ nodes and arcs takes $O(m^2)$ time because the graph is planar [4]. Since $m = O(n)$, the algorithm takes $O(n^2)$ time in total. Note that the dual arrangements have many degeneracies, because many lines can pass through the same vertex; this occurs because we added extra vertices on edges. Our definition of adjacency of faces, when they have a common edge in their boundary, handles this situation correctly.

Theorem 4 *Given a simple polygon P with n vertices, in $O(n^2)$ time we can determine if P is strictly-sweepable. That is, we can determine whether a sweep with a line ℓ over P exists such that ℓ intersects P in at most one connected component and the sweep line does not go over any point of P more than once.*

5 Conclusions and open problems

Given a simple n-vertex polyhedron in 3-dimensions, we presented an $O(n \log n)$ time algorithm to determine if the polyhedron is weakly-monotonic in the convex sense. We then presented an

$O(n^2)$ time algorithm to determine if a polyhedron is weakly-monotonic in the simple sense. Several related questions remain open. First, it is unclear whether or not $\Omega(n \log n)$ time is required to recognize if a polyhedron is weakly-monotonic in the convex sense. We believe that it should be possible to solve this problem in linear time by somehow using the fact that the input is a simple polyhedron. Next, there are many different classes of polygons that can occur in a cross-section. We only considered convex polygons and simple polygons. It would be interesting to determine if a polyhedron is weakly-monotonic in the monotonic sense (i.e. the cross-sections are monotone polygons), star-shaped sense or some other class of polygons. Furthermore, is $O(n^2)$ the best bound possible for determining if a polyhedron is weakly-monotonic in the simple sense? Can it be shown that this problem is 3-sum hard [3], or can its complexity be reduced?

We then studied variations of monotonicity in 2-dimensions, i.e. for simple polygons. Given a simple n-vertex polygon P, we showed how to determine whether or not a line ℓ can be swept over P in a continuous manner but with varying direction, such that any position of ℓ intersects P in at most two edges. We studied two variants of the problem: one where the line is allowed to sweep over a portion of the polygon multiple times (i.e. the polygon is sweepable) and one where it can sweep over any portion of the polygon only once (i.e. the polygon is strictly-sweepable). Our solutions to both problems run in $O(n^2)$ time. A natural question is whether or not this can be improved.

References

[1] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. *Comput. Aided Design*, 30(4):245–254, April 1998.

[2] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[3] A. Gajentaan and M. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory and Appl.*, 5:165–185, 1995.

[4] M. Rauch Henzinger, P. N. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55:3–23, 1997.

[5] C. Icking and R. Klein. The two guards problem. *Internat. J. Comput. Geom. Appl.*, 2(3):257–285, 1992.

[6] R. Klein. Walking an unknown street with bounded detour. *Comput. Geom. Theory Appl.*, 1:325–351, 1992.

[7] J. Milnor. *Morse Theory.* Princeton University Press, 1963.

[8] F. Preparata and K. Supowit. Testing a simple polygon for monotonicity *Information Proc. Letters*, 12(4):161–164, 1981.

[9] D. Rappaport and A. Rosenbloom. Moldable and castable polygons. *Comput. Geom. Theory and Appl.*, 4:219–233, 1994.

[10] G. T. Toussaint. Movable separability of sets. In G. T. Toussaint, editor, *Computational Geometry*, pages 335–375. North-Holland, Amsterdam, Netherlands, 1985.

[11] Handbook of Discrete and Computational Geometry (2nd Edition). J. Goodman and J. O'Rourke (editors), CRC Press, 2004.

[12] Handbook of Computational Geometry. J.-R. Sack and J. Urrutia (editors), North-Holland, 2000.