DRAWING NICE PROJECTIONS OF OBJECTS IN SPACE*

Prosenjit Bose
Department of Computer Science
University of British Columbia
Vancouver, British Columbia, Canada

Francisco Gomez
Department of Applied Mathematics
Universidad Politecnica de Madrid
Madrid, Spain

Pedro Ramos
Department of Applied Mathematics
Universidad Politecnica de Madrid
Madrid, Spain

Godfried Toussaint
School of Computer Science
McGill University
Montreal, Quebec, Canada

Abstract

Given a polygonal object (geometric graph) in three dimensional euclidean space (such as a polygon, knot or solid model mesh), we consider the problem of computing "nice" parallel projections of the object. If we imagine the viewer to be positioned at infinity above the *xy*-plane we are asked for a rotation of the object in space such that its projection on the *xy*-plane has the desired "niceness" properties. We consider a variety of definitions of "nice." One such definition, well known in the graph-drawing literature, is a projection with *few crossings*. We show that given a set of n line segments in space, deciding whether it admits a *crossing-free* projection can be done in $O(n^2 \log n + k)$ time and $O(n^2)$ space, where k is the number of such intersections and $k = O(n^4)$. Furthermore, a *minimum-crossing* projection can be found in $O(n^4)$ time and $O(n^2)$ space. Another definition of "nice" is that of a *regular* projection (of interest to knot theorists) where the projection has the property that no point of the projected image corresponds to more than two points of the original object in space. We show that a set of line segments in space (which includes polygonal objects as special cases) always admits a regular projection, and that such a projection can be obtained in $O(n^3)$ time. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k \log n)$ time, where k is the number of intersections of a set of quadratic arcs on the direction sphere and $k = O(n^6)$. Finally, when the objects are polygons and trees in space, we consider *monotonic* projections, i.e., projections such that every path from the root of the tree to every leaf is monotonic in some direction on the projection plane. We solve a variety of such problems. For example, given a polygonal chain P, we can determine in $O(n)$ time if P is monotonic on the projection plane, and in $O(n \log n)$ time we can find *all* the viewing directions with respect to which P is monotonic. In addition, in $O(n^2)$ time, we can determine all directions for which a given tree or a given simple polygon is monotonic.

* Research of the first author was supported by a Killam Fellowship. Research of the second and third authors was carried out during their visit to McGill University in 1995 and was self-supported. The fourth author was supported by NSERC Grant no. OGP0009293 and FCAR Grant no. 93-ER-0291.

1. Introduction

In our world we are frequently concerned with describing and analyzing three-dimensional (3-D) rigid objects in 3-D space. However, we often have at our disposal only a 2-D medium, such as paper or a computer-graphics screen, on which to display a necessarily incomplete representation or picture of the objects we are interested in. Therefore it is desirable to obtain 2-D representations of our objects that approximate the real objects as faithfully as possible in some sense. This family of problems, the topic of this paper, belongs to the much broader domain of *scientific visualization* [KK93], [Ga95], the discipline concerned with helping us to better visualize information, objects and processes in their fullest generality.

A sub-field of *visualization* closely related to the class of problems considered here is *graph-drawing* [DETT]. One of the archetypal problems in graph-drawing consists of asking, for a given graph, a "nice" drawing of it. A graph in this context is not a rigid object in 3-D space but a more abstract topological structure which permits the shortening, lengthening and bending of its edges to achieve the desired goal. By contrast, we are concerned with rigid *metrical* objects in 3-D space which are composed of points (vertices) and line segments (edges) and we would like to obtain "nice" *projections* of these objects on some plane that will afford them. There are many ways to project such polygonal objects onto a plane and it is therefore useful to distinguish between two fundamental and related methods: *perspective* projections and *parallel* projections [FDFH]. A perspective projection or drawing of an object represents a description of what the object looks like from a particular view point (location) in space. Obtaining perspective drawings of objects has been a fundamental visualization problem for painters for thousands of years. The first logically coherent and pictorially adequately successful methods for obtaining perspective drawings of objects appeared during the 15th and 16th centuries in the books written by artists such as Alberti in Italy, Viator in France and Dürer in Germany [Iv73].

We are concerned here on the other hand with *parallel* projections. Parallel projections may be considered as perspective projections in the limit as the view point approaches a location infinitely far away from the object being viewed. Intuitively, we may think of our object as a wire-frame sitting in 3-D space above the horizontal (xy) plane, and the parallel projection of the object on the xy-plane as the shadow cast by the wire frame when a light source shines from a point infinitely high along the positive z-axis. Obtaining "nice" parallel projections of an object then reduces to the problem of finding a suitable 3-D *rotation* for the object such that its shadow on the xy-plane contains the desired properties. Alternately, we may think of determining a direction $\boldsymbol{d}$ in space such that the projection of the object in direction $\boldsymbol{d}$ onto a plane orthogonal to $\boldsymbol{d}$ has the desired properties.

To date such problems have received scant attention in the computational geometry literature. When the objects are convex polyhedra (solid bodies) some questions have been explored. For example, a problem of interest in robotics concerns the determination of whether a convex polyhedron may be translated through a "door" that has the shape of a convex polygon. Geometrically this problem reduces to determining if the polyhedron has a shadow that fits in the door [St82], [To85]. Algorithms have also been found for determining the projections of a convex polyhedron that minimize or maximize the *area* of the shadow that the polyhedron makes on a plane when placing a light source at infinity [MS85].

In the work presented here the objects considered are polygonal structures in 3-D. Such objects include sets of disjoint line segments, 3-D simple polygons, knots, trees and, more generally, sets of segments in which the segments may touch each other at their end points. There are many specific geometrical characteristics of the vague notion of the "niceness" of a projective drawing of an object.

Some of these are more desirable than others depending on the application in mind. One requirement of "nice" is that all the significant features of the 3-D object should be visible in the projection. In other words, no vertex should lie behind another, no edge should look like a vertex and no edge should hide another edge. Furthermore, no three edges may have an interior point in common. This type of projection is useful in visualizing knots, and in knot theory is called a *regular* projection [Re83], [Li93]. Another requirement for effective visualization is *simplicity*. One measure of simplicity is the number of crossings of edges in the projection. It is desirable to obtain the projection that minimizes the number of crossings. We will refer to such projections as *minimum-crossing* projections. If the minimum number of crossings is zero we call such projections *crossing-free*. In some applications we may have a 3-D directed tree as an object of interest. Such a tree may represent a system of veins in the human brain for example, where the direction of an edge represents the direction of blood flow in the corresponding vein segment. Here it is of interest to determine if there exists a projection such that all the directions of the edges of the tree are monotonically increasing in a specified direction on the projection plane. We call such projections *monotonic* projections. In this paper we investigate these four types of projections for objects which are sets of disjoint line segments, simple polygons, polygonal chains and trees.

We should add here that the notions of minimum crossing drawings and monotonic drawings are classic visualization problems that have been well studied in the context of graph drawing [DETT]. The general question of given a graph, can one find an embedding in the plane that minimizes the number of crossing edges, is NP-complete [GJ83]. In fact this problem is also NP-complete for a variety of special cases [SSV94]. A lot of work has been done for drawing graphs in a monotonic way in the plane. These drawings are known in the graph-drawing literature as *upward planar* drawings. The general problem of determining for a given directed graph, whether it can be drawn in the plane such that every edge is monotonically increasing in the vertical direction and no two edges cross is NP-complete, as is the problem of deciding if an undirected graph can be drawn in the plane such that every edge is a horizontal or vertical segment and no two edges cross [GT95].

In this paper we consider the following problems. Given a polygonal object (geometric graph) in three dimensional euclidean space (such as a polygon, knot or solid model mesh), we consider the problem of computing "nice" parallel projections of the object. If we imagine the viewer to be positioned at infinity above the xy-plane we are asked for a rotation of the object in space such that its projection on the xy-plane has the desired "niceness" properties. We consider a variety of definitions of "nice." One such definition, well known in the graph-drawing literature, is a projection with *few crossings*. We show that given a set of n line segments in space, deciding whether it admits a *crossing-free* projection can be done in $O(n^2 \log n + k)$ time and $O(n^2)$ space, where k is the number of such intersections and $k = O(n^4)$. Furthermore, a *minimum-crossing* projection can be found in $O(n^4)$ time and $O(n^2)$ space. Another definition of "nice" is that of a *regular* projection (of interest to knot theorists) where the projection has the property that no point of the projected image corresponds to more than two points of the original object in space. We show that a set of line segments in space (which includes polygonal objects as special cases) always admits a regular projection, and that such a projection can be obtained in $O(n^3)$ time. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k \log n)$ time, where k is the number of intersections of a set of quadratic arcs on the direction sphere and $k = O(n^6)$. Finally, when the objects are polygons and trees in space, we consider *monotonic* projections, i.e., projections such that every path from the root of the tree to every leaf is monotonic in some direction on the projection plane. We solve a variety of such problems. For example, given a polygonal chain P, we can determine in $O(n)$ time if P is monotonic on the projection plane, and in $O(n \log n)$ time we can find *all* the viewing directions with respect to which

P is monotonic. In addition, in $O(n^2)$ time, we can determine all directions for which a given tree or a given simple polygon is monotonic. In this extended abstract we leave out most proofs and sketch how some results are obtained.

2. Regular projections

Let S be a set of n distinct and disjoint line segments in E^3 specified by the cartesian coordinates of their end-points (vertices of S) and let H be a plane. Let S_H be the parallel projection of S onto H. A parallel projection of S is said to be *regular* if no three points of S project to the same point on H and no vertex of S projects to the same point on H as any other point on S [Li93]. This definition implies that (1) no point of S_H corresponds to more than one vertex of S, (2) no point of S_H corresponds to a vertex of S and an interior point of an edge of S, and (3) no point of S_H corresponds to more than two interior points of edges of S. Therefore the only crossing points (intersections) allowed in a regular projection are those points that belong to the interiors of precisely two edges of S. This condition is crucial for the successful visualization and manipulation of knots [Li93]. Knots are defined as polygons in 3-D and are special cases of sets of line segments.

Given three line segments (edges of S) in E^3, all directions $\boldsymbol{d}$ that result in a non-regular projection of S in which we have a point of S_H that corresponds to three interior points of edges of S, are specified by the family of line transversals of the three edges in question. Using results by Avis and Wenger [AW87], [AW88] on transversals of three skew lines in space we show that each triple of segments of S yields an arc on the sphere of directions which corresponds to those directions that do not admit a regular projection. These arcs have measure zero on the sphere and therefore cannot cover it. In a similar way we analyze the other cases that lead to measure-zero forbidden directions to establish the following lemma.

Lemma 2.1: *A set of line segments in space always admits a regular projection.*

To compute a regular projection of a set of line segments, or a description of all the directions that admit a regular projection, we may compute the arrangement of either the $O(n^3)$ arcs on the sphere of directions or of their planar counter parts via a stereographic projection. It would be nice to be able to use the optimal algorithm of Chazelle & Edelsbrunner [CE92] for this purpose. Unfortunately, the curves in our arrangement are portions of quadratic curves and the algorithm of Chazelle & Edelsbrunner cannot handle quadratics. Recently, Amato, Goodrich and Ramos [AGR95] discovered a new optimal divide & conquer algorithm for computing the intersections of line segments that may perhaps be generalized to handle quadratics. However, at present the problem is still open. On the other hand, the original simple segment intersection algorithm of Bentley and Ottman [BO79] does handle quadratic curve segments and using these results we obtain the following.

Theorem 2.2: *Given a set of line segments in space, a regular projection can be obtained in $O(n^3)$ time. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k \log n)$ time, where k is the number of intersections of the arcs on the direction sphere and $k = O(n^6)$.*

3. Minimum-crossing projections

Whereas a regular projection of a set of line segments always exists, this is not true of crossing-free projections. To establish this it suffices to construct a counter example with three line segments very close to each other and parallel to the three orthogonal axes of the cartesian coordinate system. Here we are interested in computing a description of all the directions (if any exist) that admit crossing-free projections. Furthermore, if no crossing-free projections exist we are interested in finding projections that minimize the number of crossings. Recall that for graph-drawing problems, obtaining a minimum-crossing drawing is NP-complete [GJ83], [SSV94]. By contrast, for the projective drawing versions of these problems we provide polynomial time solutions.

Given two line segments (edges of S) in E^3, all directions $\boldsymbol{d}$ that result in a non-crossing-free projection of S in which we have a point of S_h that corresponds to two points of different edges of S, are specified by the family of line transversals of the two edges in question. In E^3 two edges of S yield a tetrahedron as a description of this family of transversals. This tetrahedron in turn determines four great-circle arcs on the sphere of directions and a resulting convex spherical quadrilateral. Thus each pair of segments of S yields a quadrilateral on the sphere corresponding to a set of directions which result in a crossing. Such quadrilaterals are termed *forbidden.* This leads to the following lemma.

Lemma 3.1: A *set of disjoint line segments in space admits a crossing-free projection iff there exists a point on the sphere of directions that it is not covered by a forbidden quadrilateral.*

The set of $O(n^2)$ forbidden spherical quadrilaterals determined by all pairs of segments in E^3 determine an arrangement on the sphere of directions. We may convert this arrangement to a suitable arrangement of straight-line (possibly unbounded) quadrilaterals in the plane via a stereographic projection. To determine if S admits a crossing-free projection then reduces to the problem of determining if the transformed quadrilaterals cover the plane. We can do this by computing the *contour of the union* of these quadrilaterals. If the contour of the union empty, then there is no direction that yields a projection without crossings.

Several algorithms have been developed for computing the contour of the union of a set of polygons. Some of these [SB92], [CN89], [NP82] are customized versions of the Bentley-Ottman line-segment intersection algorithm [BO79]. All of them compute the entire arrangement induced by the quadrilaterals and assign to each face in the arrangement, the number of quadrilaterals that intersect it. Faces numbered with zero form the contour of the union. Nievergelt & Preparata [NP82] present a version of the algorithm tailored specifically for convex polygons whose time and space complexities are $O(n \log n + k)$ and $O(n)$, respectively, where n is the number of segments in the polygons and k is the number of intersections of the segments. Souvaine & Bjorling-Sachs [SB92] propose another algorithm that computes the contour of the union from the *vertical map* by using topological sweep in time linear in the size of the map. This algorithm achieves the same time and space bounds as the algorithm of Nievergelt & Preparata [NP82]. However, in order to apply either of these algorithms to compute the contour of the union in our context, they require minor modifications to handle the unboundedness of some of our quadrilaterals. We thus establish the following theorem.

Theorem 3.2: *Given a set of n line segments in space, deciding whether it admits a crossing-free projection can be done in $O(n^2 \log n + k)$ time and $O(n^2)$ space, where k is the number of edge intersections and $k = O(n^4)$.*

If a set of line segments does not admit a crossing-free projection it is of interest to compute the projection that minimizes the number of crossings. To solve this problem we can proceed in a similar manner to that described above but this time search the entire arrangement to find the region covered with the minimum number of quadrilaterals. Therefore we have the following result.

Theorem 3.3: *Given a set of n line segments in space, a minimum-crossing projection can be found in $O(n^4)$ time and $O(n^2)$ space.*

4. Monotonic projections

Introduction

Monotonicity is another characteristic of polygonal objects that aids in their visualization. A simple polygonal chain in 3-D may not admit a crossing-free projection but it may admit a projection which is monotonic in some direction. Here we are interested in determining questions such as: does a given structure admit a monotonic projection in some unspecified direction? Such problems closely resemble the NP-complete problem of determining for a given directed graph, whether it can be drawn in the plane such that every edge is monotonically increasing in the vertical direction and no two edges cross [GT95]. Again, by contrast we provide polynomial time solutions to a variety of similar projective versions of these drawing problems.

Polygonal chains

First we study the monotonicity of polygonal chains in E^3. Specifically, we address three questions. Given a polygonal chain P and a direction $\boldsymbol{d}$, is P monotonic with respect to direction $\boldsymbol{d}$? Recall that a polygonal chain $P = v_1, v_2,..., v_n$ is *monotonic* in direction $\boldsymbol{d}$ provided that the intersection of P with every plane with normal $\boldsymbol{d}$ is empty, a point or a line segment. We show how to answer this question in $O(n)$ time, where n is the number of vertices of P. Next, given a polygonal chain P, we ask if P is monotonic in some direction? We present an algorithm that determines whether a polygonal chain is monotonic in $O(n)$ time. Finally, given a polygonal chain P, it is of interest to determine *all* directions of monotonicity of P. We show how to compute all the directions for which P is monotonic in $O(n \log n)$ time.

Given two points a and b, let $\boldsymbol{ab}$ denote the vector directed from a to b and $\boldsymbol{ba}$ the vector directed from b to a. A plane can be defined by a point p contained in that plane and the normal vector $\boldsymbol{n}$ of the plane. Given a point p and a vector $\boldsymbol{n}$, the plane defined by them is denoted by $H(p,\boldsymbol{n})$. Given a plane $h{=}H(p,\boldsymbol{n})$, we define the two half-planes determined by this plane as follows. The half-space h^+ is defined as $\{x\ /\ \boldsymbol{px}\cdot\boldsymbol{n} \geq 0\}$ and the half-space h^- is defined as $\{x\ /\ \boldsymbol{px}\cdot\boldsymbol{n} \leq 0\}$.

To avoid ambiguity and simplify the discussion, we adopt the convention that if P is monotonic in direction $\boldsymbol{d}$, then v_1 is a minimum for P with respect to $\boldsymbol{d}$. We first address the question of deciding whether a polygonal chain is monotonic in a given direction. A key property of chains monotonic with respect to $\boldsymbol{d}$ is that they can be partitioned into sub-chains and each sub-chain is also monotonic with respect to $\boldsymbol{d}$.

The above implies that it suffices to determine all directions for which a line segment is monotonic in order to compute all directions for which a polygonal chain is monotonic. A line segment is monotonic in every direction. However, by our convention, we are interested in the directions where line segment $[ab]$ is monotonic and a is minimum with respect to the given direction. The point a is a minimum with respect to all directions $D = \{\boldsymbol{d}\ /\ \boldsymbol{d}\cdot\boldsymbol{ab} \geq 0\}$. Let $h = H(O, \boldsymbol{ab})$ (where O is the origin).

It follows that all directions for which $[ab]$ is monotonic can be represented by the intersection of the half-space h^+ with the unit sphere representing all directions in space and denoted by S^2, since these are all the directions for which a is a minimum.

Let us turn to the questions introduced earlier. Given a polygonal chain $P = v_1, v_2,..., v_n$ and a direction $\boldsymbol{d}$, determine if P is monotonic with respect to $\boldsymbol{d}$. We simply need to verify that each of the line segments of P: $[v_1,v_2]$, $[v_2,v_3]$,..., $[v_{n-1},v_n]$ is monotonic with respect to $\boldsymbol{d}$, i.e., that $\mathbf{v}_j\mathbf{v}_{j+1}\cdot\boldsymbol{d} \geq 0$. We conclude with the following.

Theorem 4.1: *Given a polygonal chain P and a direction* ***d****, in O(n) time, one can determine if P is monotonic with respect to* ***d****.*

We can determine if a polygonal chain $P = v_1, v_2,..., v_n$ is monotonic for some direction in the following way. Let h_i represent the plane $H(O, \boldsymbol{v}_i\boldsymbol{v}_{i+1}\})$. Let $D = \bigcap_{i=1}^{n} h_i$. The set of all directions for which P is monotonic is described by $D \cap S^2$. Now the intersection $D \cap S^2$ is not empty when D is non-trivial (i.e. not the origin O). Note that the origin is contained in D since all planes h_i contain the origin. Determining if D, the intersection of a set of half-spaces, is non-trivial, can be accomplished in linear time using linear programming [Me83]. Therefore we conclude with the following.

Theorem 4.2: *Given a polygonal chain P, one can determine if P is monotonic in O(n) time.*

As noted above, $D \cap S^2$ describes the set of all the directions from which P is monotonic. Since the intersection of a set of half-spaces can be computed in $O(n \log n)$ time, we conclude with the following.

Theorem 4.3: *Given a polygonal chain P, one can determine in O(n log n) time all the directions with respect to which P is monotonic.*

Projections of monotonic trees and polygons

In this sub-section, we ask the same questions as in the previous section, however, the objects of study are more complex. We study the monotonicity of simple polygons and trees in E^3.The polygonal chains, simple polygons and trees in E^3 are all graphs embedded in E^3. In order to continue the discussion in this more general setting, we define a *geometric graph*. A *geometric graph* is a two-tuple (V,E), where V is a finite set of distinct points in general position in E^3, and E is a family of closed straight-line segments with end-points in V. The elements of V and E are called *vertices* and *edges*, respectively. For more definitions and terminology concerning graphs, the reader is referred to [BM76] or [Ha72]. In the previous section, the geometric graphs that we considered were paths. In this section, we concentrate on trees and cycles (polygons).

We begin by describing some properties of geometric graphs. Given a vertex v of a geometric graph G, we denote the set of **e**dges **a**djacent to v by $EA(v)$.

Lemma 4.4: *Vertex v is a minimum with respect to* ***d*** *for EA(v) if and only if* $\forall\ e \in EA(v)$*, v is a minimum with respect to* ***d*** *for e.*

Proof: ($\Rightarrow$) Let $h = H(v,\boldsymbol{d})$. Since v is a minimum for $EA(v)$, it follows that $EA(v)$ must be contained in h^+. This implies that each of the edges $e \in EA(v)$ is contained in h^+.

($\Leftarrow$) Let $h = H(v,\boldsymbol{d})$. Since v is a minimum $\forall\ e \in EA(v)$, each e is contained in h^+. But this implies that $EA(v)$ is contained in h^+.

Given a vertex v of a geometric graph G, we denote by $MD(v)$ the set of directions for which v is a minimum for the set $EA(v)$. Let $e = [vv_i]$ be an edge in $EA(v)$. By $\boldsymbol{e}$ we denote the vector $v\,v_i$. Let $h(e) = H(O,\boldsymbol{e})$. We see that

$$MD(v) = \bigcap_{\forall e \in EA(v)} h(e)$$

Definition: A vertex v of a geometric graph is a local minimum with respect to direction $\boldsymbol{d}$ provided that v is a minimum for the set $EA(v)$ in direction $\boldsymbol{d}$.

We now address several questions concerning the monotonicity of trees. Suppose we are given a rooted tree T, and a direction $\boldsymbol{d}$. The first question we address is to determine if T is monotonic in direction $\boldsymbol{d}$. Notice that two things are specified in this question, the *root* of the tree and the proposed *direction* of monotonicity. The next four questions we address are the following:

(1) Given a rooted tree T, does there exist a direction $\boldsymbol{d}$ for which T is monotonic?

(2) Given an unrooted tree T and direction $\boldsymbol{d}$, does there exist a root such that T is monotonic with respect to $\boldsymbol{d}$?

(3) Given an unrooted tree T, does there exist a direction $\boldsymbol{d}$ and a root of T such that T is monotonic with respect $\boldsymbol{d}$?

(4) Given an unrooted tree T, find all roots and directions for which T is monotonic.

Recall that a tree T is called a *rooted tree* if a unique vertex v of T is specified to be the root, otherwise the tree is *unrooted* or *free*. A rooted tree T is monotonic in direction $\boldsymbol{d}$ provided that the path from the root to every vertex is monotonic in direction $\boldsymbol{d}$.

The key behind the efficient solution of all above-mentioned problems depends on the following characterization of the monotonicity of rooted trees.

Lemma 4.5: *A rooted tree T is monotonic in direction* **d** *if and only if the root r of T is the only local minimum with respect to direction* **d***.*

Proof: ($\Rightarrow$) Assume T is monotonic with respect to $\boldsymbol{d}$. Suppose there exists a vertex v of T that is not the root, such that v is a local minimum. Let $h = H(v,\boldsymbol{d})$. By Lemma 4.4, $EA(v) \in h^+$. Let P be the unique path from r to v in T. P must be monotonic since T is monotonic. Also, the root r is a minimum and v is a maximum for P with respect to direction $\boldsymbol{d}$, by the convention of monotonic paths. Therefore, P must be contained in h^-. Let v_j be the vertex preceding v in P (v_j might be the root). Since v_j is adjacent to v, we see that $[vv_j] \in EA(v)$. But the fact that $[vv_j] \subset h^-$ contradicts the assumption that v is a local minimum.

($\Leftarrow$) Given a vertex v, we will represent the plane $H(v,\boldsymbol{d})$ by h_v. Assume that the root r of T is the only local minimum. Let v be an arbitrary vertex of T. We will show that the path P from r to v must be monotonic in direction $\boldsymbol{d}$. Let the path $P = v_1\,(=r), v_2, \ldots, v_{k-1}, v_k\,(=v)$. Suppose P is not monotonic with respect to $\boldsymbol{d}$. Let v_i be the first vertex of P such that $[v_{i-1}v_i]$ is not monotonic with respect to $\boldsymbol{d}$. Since $[v_{i-1}\ v_i]$ is not monotonic with respect to $\boldsymbol{d}$, we conclude that v_i must be contained in $h^-_{v_{i-1}}$. Since v_i is not a local minimum, and T is acyclic, there must exist a vertex v_j in $EA(v_i)$ which is not the root, such that v_j is contained in $h^-_{v_i}$. Similarly, since v_j is not a local minimum, there must exist a v_m in $EA(v_j)$ which is not the root, such that v_m is contained in $h^-_{v_j}$. By continuing this argument, it follows that since there are only a finite number of vertices in T, there must be a vertex v_t such that $EA(v_t)$ is contained in $h^+_{v_t}$, contradicting the fact that r is the only local minimum.

Suppose we are given a tree T, a root r of T and a direction $\boldsymbol{d}$ and want to determine whether T is monotonic with respect to $\boldsymbol{d}$. By Lemma 4.5, since r is the only local minimum with respect to $\boldsymbol{d}$, the direction $\boldsymbol{d}$ cannot be contained in $MD(v)$ for all vertices of T other than the root. Since $MD(v)$ is an intersection of half-spaces, determining if the direction $\boldsymbol{d}$ is contained in $MD(v)$ can be done in $O(|M(v)|)$ time using linear programming [Me83]. But $O(|M(v)|)$ is $O(d(v))$ where $d(v)$ is the degree of v in the tree T. Therefore, since the sum of the degrees of the vertices of a tree is linear in the number of vertices of the tree, we conclude that in $O(n)$ time where n is the number of vertices of T, we can determine if a rooted tree is monotonic in a given direction $\boldsymbol{d}$.

Theorem 4.6: *Given a rooted tree T, and a direction* $\boldsymbol{d}$*, one can decide in* $O(n)$ *time if T is monotonic with respect to* $\boldsymbol{d}$*.*

Suppose next that the root of the tree is no longer specified. Then, to determine if T is monotonic in direction $\boldsymbol{d}$, we must first find a root. By Lemma 4.5, the root must be the only local minimum. Therefore, there must exist exactly one vertex r of T such that $\boldsymbol{d}$ is contained in $MD(r)$, which becomes the root. All other vertices v of T must have the property that $\boldsymbol{d}$ is not in $MD(v)$ Again this can be determined in linear time. Therefore, we conclude with the following.

Theorem 4.7: *Given an unrooted tree T, and a direction d, one can decide in* $O(n)$ *time where n is the number of vertices of T, if there exists a root r of T such that T is monotonic with respect to d.*

Before continuing, we need a few more preliminaries. Instead of representing all directions in 3-space by the sphere of directions S^2, we will represent all directions in E^3 by points on the surface of the axis-parallel cube AC centered at the origin O and with edge length 2. A point p on AC represents the direction $\boldsymbol{Op}$. Although this representation is not standard, it will simplify many of the algorithms to follow.

Now, suppose that the root r of T is specified but the direction is not. For T to be monotonic in some direction $\boldsymbol{d}$, we see that $\boldsymbol{d}$ must be in $MD(r)$ but outside $MD(v)$ for all vertices v of T. This problem, is in fact as difficult as the general problem where given an unrooted tree T, find all possible roots and directions for which T is monotonic. As such we will present the solution to the general problem below.

The intersection of $MD(v)$ and AC represents the set of directions for which v is a local minimum. Since $MD(v)$ is the intersection of a set of half-spaces, the intersection of $MD(v)$ with a facet F of AC is either empty, the facet itself, or a convex polygon. For each facet F_i of AC $(1 \leq i \leq 6)$ and each vertex v_j of T $(1 \leq j \leq n)$, we compute the intersection $I(i,j) = MD(v_j) \cap F_i$

On each facet F_i, notice that the set $I_i = \{I(i,j) / 1 \leq j \leq n\}$ is simply a collection of convex polygons. This collection of polygons has the following property. If a point $p \in F_i$ is contained in k polygons of the set I_i, then there are k vertices of T that are local minima with respect to the direction $\boldsymbol{Op}$ and each of the vertices that are local minima is identified by the polygon which contains p. That is, if p is contained in polygon $I(i,3)$ then vertex v_3 is a local minimum with respect to direction $\boldsymbol{Op}$.

Therefore, to determine if there are any directions with respect to which T is monotonic, we want to determine if there are any regions in each facet F_i ($1 \le i \le 6$) that are covered by only one polygon of the set I_i ($1 \le i \le 6$). In fact, we want to find all regions that are covered by only one polygon. This set of regions represents the set of all the directions and roots from which T is monotonic.

Let A_i represent the subdivision induced on facet F_i by the set of polygons I_i. This subdivision can be computed deterministically in $O(n \log n + t)$ time where t is the total number of intersection points of all polygons in I_i. The complexity of A_i is $O(n^2)$. Consider the graph G_i which has a node for every cell of A_i, and an edge between two nodes if the corresponding cells are incident to the same edge of A_i. The graph G_i has $O(n^2)$ nodes and edges. Start at any node a_1 of G_i and compute in $O(n)$ time how many polygons of I_i cover it. Store this number with a_1. Start from a_1 with a depth first search. Every edge (a_1, a_m) of G_i we traverse corresponds to going inside or outside a polygon of I_i, in which case we take the number of a_1 and add or subtract one from it and assign this number to a_m. Thus the whole process of assigning values to nodes of G_i can be done in $O(n^2)$ time. Let M_i represent the set of nodes with the minimum number assigned to it. If this number is one, then each of the cells represented by a node in M_i represents a set of directions and a root from which T is monotonic. The root of T is specified by the vertex generating the convex polygon covering the cell.

Theorem 4.8: *In $O(n^2)$ time, we can determine all directions for which T is monotonic.*

Consider now the problem of determining the monotonicity of a simple polygon in E^3. We begin with a few definitions. A simple polygon in E^3 is a geometric graph that is a cycle. A simple polygon P is monotonic in direction $\boldsymbol{d}$ provided there exist two vertices u,v of P such that both paths from u to v are monotonic in direction $\boldsymbol{d}$.

Lemma 4.9: *A simple polygon P is monotonic in direction **d** if and only if it has only one local minimum with respect to direction **d**.*

The characterization of monotonicity of simple polygons in E^3 is similar to that of trees and therefore the solution for trees is applicable in this case. Therefore, we conclude with the following.

Theorem 4.10: *Given a simple polygon P in E^3 and a direction **d**, in $O(n^2)$ time it can be determined if P is monotonic with respect to **d**.*

Theorem 4.11: *Given a simple polygon P in E^3, in $O(n^2)$ time, we can determine all the directions with respect to which P is monotonic.*

5. Conclusion

We have initiated the study of "nice" projections of polygonal structures (or geometric graphs) in 3-D. Our results on regular and minimum-crossing projections of line segments have immediate corollaries for polygonal chains, polygons, trees and more general geometric graphs in 3-D since these are all special cases of sets of line segments. All the questions we have addressed from the parallel projection point of view have their perspective projection counter parts. We are presently investigating these and related questions.

6. Acknowledgments

The authors thank Suneeta Ramaswami and Fady Habra for discussions on these problems in a computational geometry seminar at McGill University where these problems were posed. We also thank Bernard Chazelle and Mike Goodrich for correspondence on the computation of the intersections of quadratic curves. The second author thanks Lola for her support.

7. References

[AGR95] Amato, N. M., Goodrich, M. T. and Ramos, E. A., "Computing faces in segment and simplex arrangements," *Proc. Symposium on the Theory of Computing*, 1995.

[AW88] Avis, D., and Wenger, R., "Polyhedral line transversals in space," *Discrete and Computational Geometry*, vol. 3, 1988, pp. 257-265.

[AW87] Avis, D., and Wenger, R., "Algorithms for line stabbers in space," *Proc. Third ACM Symp. on Computational Geometry*, 1987, pp. 300-307.

[BO79] Bentley, J.L. and Ottmann, T.A., "Algorithms for reporting and counting geometric intersections", *IEEE Trans. Comput.* vol. 8, pp. 643-647, 1979.

[CE92] Chazelle, B. and Edelsbrunner, H., "An optimal algorithm for intersecting line segments in the plane," *J. ACM*, vol. 39, 1992, pp. 1-54.

[CN89] Chiang, K., Nahar, S., and Lo, C., "Time-efficient VLSI artwork analysis algorithms in GOALIE2", *IEE Trans. Computer-Aided Design*, vol. 39, pp. 640-647, 1989.

[DETT] Di Battista, G., Eades, P., Tamassia, R. and Tollis, I. G., "Algorithms for drawing graphs: an annotated bibliography," *Computational Geometry: Theory and Applications*, vol. 4, 1994, pp. 235-282.

[FDFH] Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F., *Computer Graphics: Principles and Practice*, Addison-Wesley, 1990.

[Ga95] Gallagher, R. S., Ed., *Computer Visualization: Graphic Techniques for Engineering and Scientific Analysis*, IEEE Computer Society Press, 1995.

[GJ83] Garey, M.R., and Johnson, D.S., "Crossing number is NP-complete," *SIAM J. Alg. Discrete Methods*, vol. 4, 1983, pp. 312-316.

[GT95] Garg, A. and Tamassia, R., "On the computational complexity of upward and rectilinear planarity testing," eds., R. Tamassia and I. G. Tollis, *Proc. Graph Drawing'94*, *Lecture Notes in Computer Science*, vol. 894, Springer-Verlag, 1995, 286-297.

[Iv73] Ivins, W. M., Jr., *On the Rationalization of Sight*, Da Capo Press, Inc., New York, 1973.

[KK93] Keller, P. R. & Keller, M. M., *Visual Cues: Practical Data Visualization*, IEEE Computer Society Press, 1993.

[Li93] Livingston, C., *Knot Theory*, The Carus Mathematical Monographs, vol. 24, The Mathematical Association of America, 1993.

[Me83] Megiddo, N., "Linear-time algorithms for linear programming in R^3 and related problems," *SIAM Journal of Computing*, vol. 12, 1983, pp. 759-776.

[MS85] McKenna, M. & Seidel, R., "Finding the optimal shadows of a convex polytope," *Proc.*

ACM Symposium on Computational Geometry, Baltimore, 5-7, June 1985, pp. 24-28.

[NP82] Nievergelt, J., Preparata, F., "Plane-sweep algorithms for intersecting geometric figures", *Communications of ACM* vol.25, pp. 739-747, 1982.

[PS85] Preparata, F., and Shamos, M., *Computational Geometry: An introduction*, Springer-Verlag, New York, 1985.

[Re83] Reidemeister, R., *Knot Theory*, BSC Associates, Moscow, Idaho, USA, 1983.

[SB92] Souvaine, D. and Bjorling-Sachs, I., "The contour problem for restricted-orientation polygons", *Proc. of the IEEE*, vol. 80, pp. 1449-1470, 1992.

[SSV94] Shahrokhi, F., Szekely, L., and Vrt'o, I., "Crossing number of graphs, lower bound techniques and algorithms: A survey," *Lecture Notes in Comp. Science, vol. 894*, Princeton, New Jersey, 1994, pp. 131-142.

[St82] Strang, G., "The width of a chair," *The American Mathematical Monthly*, vol. 89, No. 8, October 1982, pp. 529-534.

[To85] Toussaint, G. T., "Movable separability of sets," in *Computational Geometry*, G. T. Toussaint, Ed., Elsevier Science Publishers, 1985, pp. 335-375.