Drawing Nice Projections of Objects in Space[†]

Prosenjit Bose [‡] *Francisco Gómez* [§] *Pedro Ramos* [¶] *Godfried Toussaint*[‖]

Abstract

Given a polygonal object (simple polygon, geometric graph, wire-frame, skeleton or more generally a set of line segments) in three dimensional Euclidean space, we consider the problem of computing a variety of "nice" parallel (orthographic) projections of the object. We show that given a general polygonal object consisting 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 $O(n^2 + k)$ space, where k is the number of edge intersections of forbidden quadrilaterals (i.e. set of directions that admits a crossing) and varies from zero to $O(n^4)$. This implies for example that given a simple polygon in 3-space we can determine if there exists a plane on which the projection is a simple polygon, within the same complexity. Furthermore, if such a projection does not exist, a *minimum-crossing* projection can be found in $O(n^4)$ time and space. We show that an object always admits a regular projection (of interest to knot theory), and that such a projection can be obtained in $O(n^2)$ time and space or in $O(n^3)$ time and linear space. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k)$ time , where k is the number of intersections of a set of quadratic arcs on the direction sphere and varies from $O(n^3)$ to $O(n^6)$. 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 a common 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 with respect to which a given tree or simple polygon is monotonic.

[†]Research of the first author was supported by NSERC Grant OGP0183877. 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.

[‡]School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6 (jit@scs.carleton.ca).

[§]Escuela Universitaria de Informática, U.P.M., Madrid, España (fmartin@eui.upm.es)

[¶]Facultad de Informática, U.P.M., Madrid, España (pramos@fi.upm.es)

[‖]School of Computer Science, McGill University, Montreal, Quebec, Canada (godfried@cs.mcgill.ca)

1 Introduction

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 [Kel93], [Gal95]. A sub-field of *visualization* closely related to the class of problems considered here is *graph-drawing* [DBETT94]. 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.

We are concerned here with *parallel* (orthogonal) projections [FvDFH90] rather than perspective 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.

To date such problems have received scant attention in the computational geometry literature. When the objects are convex polyhedra (solid bodies) several questions have been explored. For example, a problem of interest in robotics concerns the determination of whether a convex polyhedron P 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 [Str82], [Tou85]. 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], [BGK95]. In computer graphics, good projections for radiosity computation are those that yield the most number of facets visible from the viewpoint [Col90]. On the other hand, when the objects are 3-D polygonal objects (skeletons or wire-frames) very little is known. Hirata et al., [HMTT94] give bounds on the worst-case combinatorial complexity of the simplest projections of the skeletons of 3-D convex subdivisions onto a plane. Such simple projections have application to the design of efficient 3-D point location query algorithms [PT92]. Closer in spirit to the work presented here, Kamada and Kawai [KK88] present an $O(n^6 \log n)$ time algorithm for computing the projection of a wire-frame, that in a sense maximizes the projected minimum distance between parallel segments. Finally, Bhattacharya and Rosenfeld [BR94] have studied a special class of orthographic projections called Wirtinger projections for 3-D polygons. Independent of this work, Barequet et al.[BDE96] studied orthographic projections for the special case of simple polygons.

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, such as skeletons of 3-D Voronoi diagrams or other subdivisions such as those in [HMTT94]. 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, closely related to Wirtinger projections [BR94], is useful in visualizing knots, and in knot theory is called a *regular* projection [Rei32], [Liv93]. 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. In general we call such projections *monotonic* projections. More specifically, a projection is monotonic if the projected image on the projection plane is monotonic. A planar polygonal chain is monotonic if there exists a direction such that every line orthogonal to this direction, that intersects the chain, yields a point as the intersection. A planar polygon is monotonic if it can be partitioned into two chains each of which is monotonic with respect to the same direction. A tree is monotonic if it contains a root and a direction such that all paths from the root to the leaves are monotonic with respect to that direction. In this paper we investigate the above 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 [DBETT94]. 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. Computing an upward planar drawing of a graph refers to 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. This problem 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 most general polygonal object, i.e., a set of disjoint line segments. 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 $O(n^2 + k)$ space, where k is the number of edge intersections of forbidden quadrilaterals (i.e. set of directions that admits a crossing) and $k = O(n^4)$. This implies for example that given a simple polygon in 3-space we can determine if there exists a plane on which the projection is a simple polygon, within the same complexity. Furthermore, if such a projection does not exist, a *minimum-crossing* projection can be found in $O(n^4)$ time and space.

We show that a set of line segments in space (which includes polygonal objects as a special case) always admits a regular projection (of interest to knot theory), and that such a projection can be obtained in $O(n^2)$ time and space or in $O(n^3)$ time and linear space. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k)$ time , where k is the number of intersections of a set of quadratic arcs on the direction sphere and varies from $O(n^3)$ to $O(n^6)$. 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 $k = O(n)$ 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 with respect to which a given tree or a given simple polygon is monotonic.

2 Regular and Wirtinger 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 [Liv93]. This definition implies that for disjoint line segments (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 [Liv93]. Knots are defined as polygons in 3-D and are special cases of sets of line segments where not all segments are disjoint. Note that a vertex where two edges are joined together in the case when the line segments form a 3-D polygon counts as (not two but) one vertex. Regular projections of 3-D polygons were first studied by the knot theorist K. Reidemeister in 1932 [Rei32] who showed that all 3-D polygons admit a regular projection and in fact almost all projections of polygons are regular. This result was re-discovered by Bhattacharya and Rosenfeld [BR94] for a restricted class of regular projections known as *Wirtinger* projections. Regular projections allow two consecutive edges of a 3-D polygon to project to two collinear consecutive edges on H. Therefore some shape features of the polygon are lost in regular projections. For visualization applications this may not be desirable. Those regular projections in which it is also required that no two consecutive edges of the 3-D polygon have collinear projections, are known as *Wirtinger* projections. The above authors did not address the algorithmic complexity of actually finding regular or Wirtinger projections. In this section we study the complexity of computing a single regular or Wirtinger projection as well as constructing a description of all such projections for the more general input consisting of disjoint line segments. These results include therefore results for 3-D chains, polygons, trees and geometric graphs in general. The description of all projections allows us to obtain regular or Wirtinger projections that optimize additional properties. For example, one may be interested in obtaining the most tolerant projection in the sense that it maximizes the deviation of the view-point required to violate the regularity property. We begin by showing that every set of disjoint line segments always admits a regular projection.

Lemma 1 *A set of line segments in space always admits a regular projection.*

Proof: We first consider the case of a point in the projection coming from three or more points of S. A line that intersects three or more line segments determines a forbidden direction of projection. These lines constitute the family of transversals of three line segments. Let us consider the lines going through the line segments of S. The set of transversals of any three of these lines forms:

1. a *hyperboloid of one sheet* when the lines are skew and not parallel to a common plane,
2. a *hyperbolic paraboloid* when the lines are skew and parallel to a common plane,
3. *two planes* if two of the lines meet at a point,
4. a *plane* if two of the lines are coplanar and the third one intersects the plane.

Transversals are well-studied objects and the reader is referred to [Bor69], [HCV83], or [AW88] for proofs of the above facts. Any of the three surfaces is ruled and generated by a uni-parametric family of lines. Let us analyze the structure of the set of directions given by these surfaces:

1. If the surface is a hyperboloid of one sheet, then its canonical equation is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \quad a, b, c \neq 0. \tag{1}$$

   Substituting the following,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = \left(\frac{x}{a}\cos(t) + \frac{y}{b}\sin(t)\right)^2 + \left(\frac{x}{a}\sin(t) - \frac{y}{b}\cos(t)\right)^2,$$

   into Eq. (1) yields

$$\left(\frac{x}{a}\cos(t)+\frac{y}{b}\sin(t)\right)^2+\left(\frac{x}{a}\sin(t)-\frac{y}{b}\cos(t)\right)^2-\frac{z^2}{c^2}=1 \implies$$
$$\left(\frac{x}{a}\cos(t)+\frac{y}{b}\sin(t)\right)^2-\frac{z^2}{c^2}=1-\left(\frac{x}{a}\sin(t)-\frac{y}{b}\cos(t)\right)^2 \implies$$
$$\implies \begin{cases} \dfrac{x}{a}\sin(t)-\dfrac{y}{b}\cos(t)=1 \\[2ex] \dfrac{x}{a}\cos(t)+\dfrac{y}{b}\sin(t)-\dfrac{z}{c}=0. \end{cases} \tag{2}$$

Eq. (2) yields two planes whose intersection is a line contained in the hyperboloid of one sheet. As t varies in the interval $[0, 2\pi]$, the uni-parametric family of lines is generated. The cross product of the normal vectors of the planes given by Eq. (2) produces the directions of the generating lines.

$$\left(\frac{1}{a}\cos(t),\frac{1}{b}\sin(t),\frac{-1}{c}\right)\times\left(\frac{1}{a}\sin(t),\frac{-1}{b}\cos(t),0\right)=\left(\frac{1}{bc}\cos(t),\frac{1}{ac}\sin(t),\frac{1}{ab}\right).$$

As can be seen, the above set of directions determines an elliptic cone.

2. If the surface is a hyperbolic paraboloid, then its canonical equation is:

$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=2z \quad a,b\neq 0. \tag{3}$$

If we set

$$\frac{x}{a}-\frac{y}{b}=t, t\neq 0.$$

and substitute this expression into Eq. (3), then the following two planes are obtained:

$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=2z \implies \left(\frac{x}{a}-\frac{y}{b}\right)\cdot\left(\frac{x}{a}+\frac{y}{b}\right)=2z \implies$$
$$\implies \begin{cases} \dfrac{x}{a}-\dfrac{y}{b}=t \\[2ex] \dfrac{x}{a}+\dfrac{y}{b}-\dfrac{2z}{t}=0. \end{cases} \tag{4}$$

Again, they intersect each other giving rise to the corresponding family of lines whose directions are contained in a plane and given by:

$$\left(\frac{1}{a},\frac{-1}{b},0\right)\times\left(\frac{1}{a},\frac{-1}{b},\frac{-2}{t}\right)=\left(\frac{2}{bt},\frac{2}{at},\frac{2}{ab}\right).$$

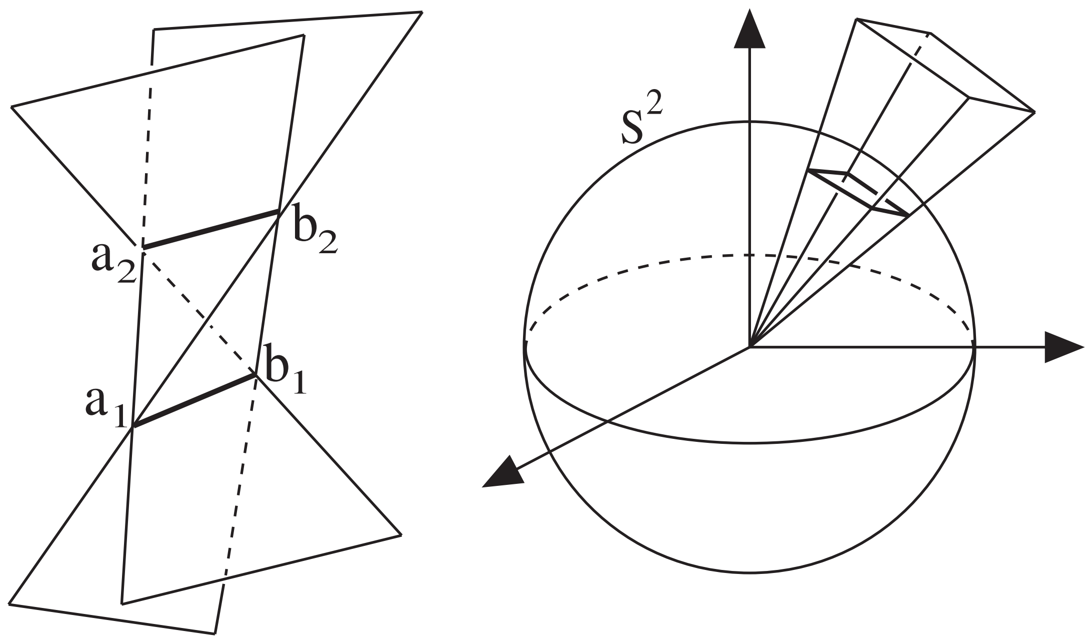

Figure 1: Forbidden quadrilaterals.

3. If the surface is a plane, the generating lines are constructed by taking all the lines contained in a plane through a given point.

The directions specified by these lines, when represented on the sphere of directions S^2, give rise to either ellipses or great circles. Since we restricted ourselves to line segments instead of lines, the forbidden directions consist of arcs of ellipses or great circles.

We now turn our attention to the case in which a vertex of S is projected onto another line segment (cases 2 and 3 of the definition). The forbidden directions are formed by the line transversals of a vertex and a line segment. Let $s_1 = a_1b_1$ and $s_2 = a_2b_2$ be two segments in S. The directions that project a_1 on s_2, for example, are given by the lines going through a_1 and any point of s_2. Those lines are contained in the triangular wedge determined by a_1, b_2 and a_2 (refer to Figure 1). For each vertex, the line segment that does not contain it determines three other wedges. On the sphere of directions, such wedges produce a quadrilateral composed of arcs of great circles. Finally, a vertex can be projected onto the line segment to which it belongs. The line through the segment is the only transversal and a forbidden point results on the sphere.

The forbidden directions are composed of a finite number of arcs of measure zero on the sphere and therefore cannot cover it.

■

To compute a regular projection of a set of line segments, or a description of all the directions that admit a regular projection, one may in theory compute the arrangement of the $O(n^3)$ arcs on the sphere. However, the intersection of two quadratic surfaces yields arcs on the sphere that are space curves of degree four and computing the arrangement of such curves is difficult in practice. A much better approach is to project these arcs from the sphere to the plane $z = 1$ since then we only need to compute the arrangement of a set of quadratic arcs on the plane (see Harris [H92] for a proof).

Theorem 1 *Given a set of line segments in space, a regular projection can be obtained in $O(n^3)$ time and linear space.*

Proof: To compute a direction of regular projection we need to choose a point on the plane of directions such that it neither belongs to an edge of the arrangement nor is a forbidden point. Here we present an algorithm for computing a set of regular directions (Refer to Figure 2).

1. Form the set L of all forbidden arcs of directions that go through the north pole N and are given by two segments of S. If L is empty, pick any point Q on the plane $z = 1$ and go to step 3.

2. Find two consecutive elements in L with distinct slopes s_1 and s_2. Construct a segment NQ with slope between s_1 and s_2.

3. Intersect all forbidden directions not in L with NQ. Among all intersection points, pick the point closest to N. Let R be that point.

4. Exit with NR as an open segment of regular directions.

We turn to the correctness of the algorithm. Suppose that NR contained a point p giving a non-regular projection. Such a point p could not be an isolated point. If it were, it would come from an intersection of forbidden points or quadratic curves or segments not parallel to NR. However, step 3 of the algorithm rules out such a possibility. Therefore, point p cannot be isolated and there must exist a line segment of forbidden directions contained in NR. That segment must contain N since in step 3, again, segments not going through N were disregarded. This contradicts the fact that NR has a different slope from any segment going through N.

The set L need not be actually constructed; it is sufficient to enumerate its elements. Therefore the above algorithm uses only linear space.

■

However, the time complexity of this algorithm can be reduced from $O(n^3)$ to $O(n^2)$ at the expense of increasing the space complexity from $O(n)$ to $O(n^2)$, as we now demonstrate.

Theorem 2 *Given a set of line segments in space, a regular projection can be obtained in $O(n^2)$ time and space.*

Proof: As in the above algorithm, we first compute segment NQ and next obtain all the intersection points between NQ and all the curves of degree one not going through N.

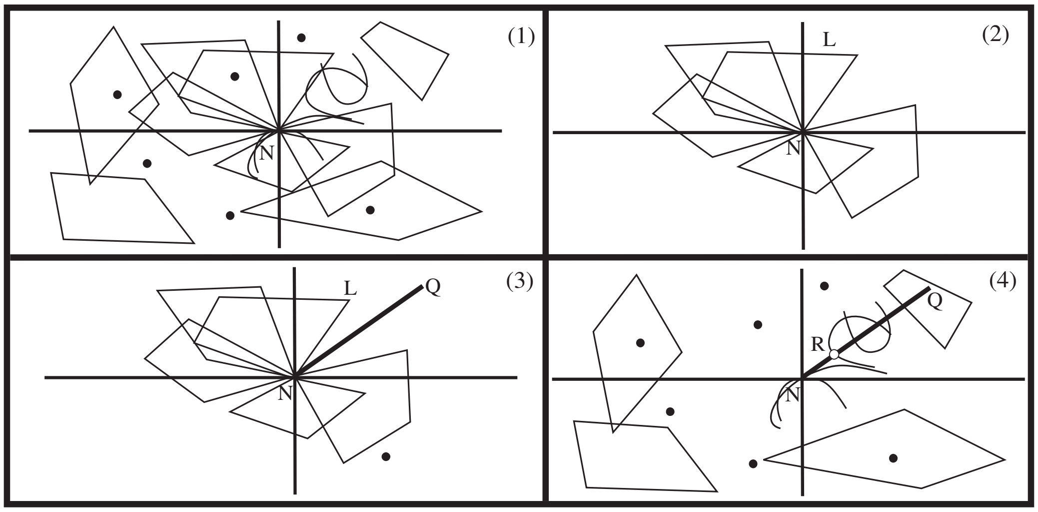

Figure 2: Computing a regular direction in $O(n^3)$ time and linear space.

Let Q' be the intersection point closest to N. The segment NQ' is not necessarily free of forbidden directions since the quadratic curves have not been taken into account yet. A quadratic curve which goes through N certainly has to be considered since it could intersect NQ'. Now let us compute the projection of S onto the xy-plane and its corresponding arrangement $A(S)$ (refer to Figure 3). All quadratic curves going through N are determined by the $O(n^2)$ concurrent segments in the arrangement $A(S)$. Let us intersect such quadratic curves with NQ' so a new line segment NQ'' is obtained. From the remaining quadratic curves, only those given by the segments of triangular faces in the arrangement $A(S)$ have to be considered. Indeed, if we move along segment NQ'', then the first intersection point encountered with a curve must come from three segments forming a triangular face. Suppose this is not the case. Then three segments that did not form a triangular face would become concurrent. However, for this to happen, at some point one of their vertices must have been contained in another segment. This is a contradiction since such a case has already been ruled out by the construction of NQ''. Finally, we intersect all the quadratic curves determined by the triangular faces in $A(S)$ and compute the point R closest to the north pole N. Segment NR is the desired open segment of regular directions.

Since the number of triangular faces in $A(S)$ is $O(n^2)$, the total time complexity is also $O(n^2)$. As for the space complexity, the arrangement $A(S)$ can be stored using $O(n^2)$ space. ■

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

Proof: In order to compute a description of all regular directions we could use several of the existing optimal segment-intersection algorithms for computing the intersection of

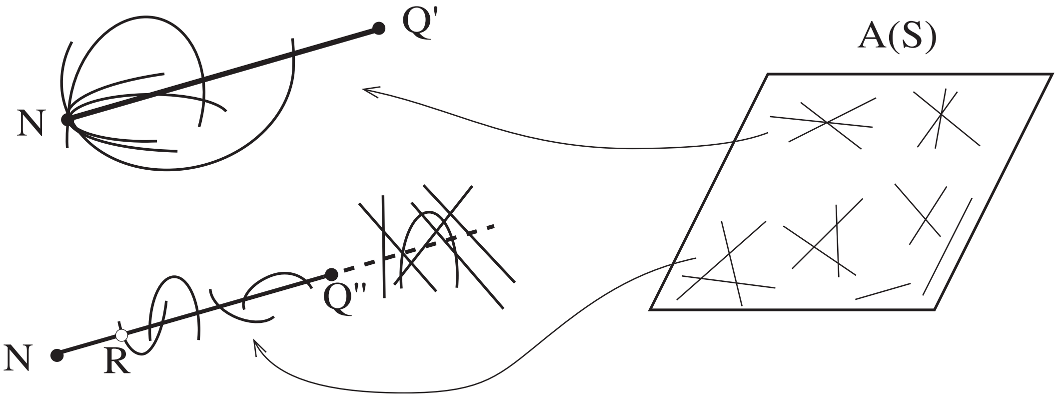

Figure 3: Computing a regular direction in $O(n^2)$ time and space.

a set of arcs on the plane. The algorithms of Chazelle & Edelsbrunner [CE92] or Amato, Goodrich and Ramos [AGR95] do not appear to be able to be modified to handle quadratic curve segments. However, recently Balaban [Bal95] discovered an optimal algorithm that computes all intersections of quite general curves, including quadratics, that has time and space complexities $O(n \log n + k)$ and $O(n)$, respectively, where k is the number of intersections among the curves. By using his algorithm, we achieve the desired time and space complexities.

■

One may wonder if it is worth using the optimal quadratic curve segment intersection algorithm of Balaban in practice given that there is a suboptimal but very simple algorithm due to Bentley and Ottman [BO79] that also handles quadratic curve segments and has time and space complexities $O(n \log n + k \log n)$ and $O(n)$, respectively, where k is the number of intersections among the curves. Balaban has conducted experiments comparing his optimal algorithm to the Bentley-Ottman algorithm for as many as $4,000$ segments and the latter algorithm was twice as fast. In fact, Balaban suggests that in practice the suboptimal algorithm should be used unless the number of segments is at least $200,000$.

Recall that a Wirtinger projection [BR94] of a 3-D polygon is a special type of regular projection in which no two adjacent edges project to a pair of collinear edges. We can use the above approach to compute Wirtinger projections of polygons also.

Theorem 4 *Given a polygon P in space, a Wirtinger projection of P can be obtained in $O(n^3)$ time and $O(n)$ space. A description of the set of all directions which yield a Wirtinger projection of P can be computed in $O(n^3 \log n + k)$ time, where k is the number of intersections of arcs and great circles on the direction sphere and $k = O(n^6)$*

Proof: For Wirtinger projections we have, in addition to the $O(n^3)$ forbidden curve segments on the direction sphere, a set of n additional forbidden great circles. Each pair of adjacent edges of the 3-D polygon yields a plane that contains them. Translate this plane

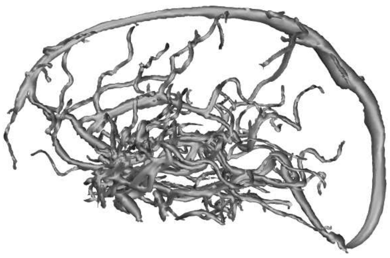

Figure 4: A side view of the veins in the brain.

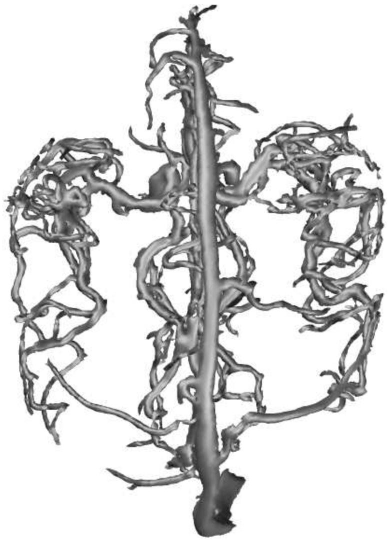

Figure 5: A top view of veins in the brain.

to the origin and intersect it with the sphere of directions. This intersection is a forbidden great circle of directions since for each view point on this circle the two adjacent edges appear to be collinear. In total we still have $O(n^3)$ forbidden curve segments and great circles.

■

3 Minimum-crossing projections

A projection with many crossings is more difficult to visualize than one with few crossings. Consider the Figures 4 and 5 that show different projections of the veins in the human brain. The projection in Figure 5 has fewer crossings than the other. Therefore, it is of interest to compute 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. Minimum-crossing projections are also of interest in knot theory. The number of crossings in the projection with the minimum number of crossings is called the *crossing number* of a knot. Recall that for graph-drawing problems, obtaining a minimum-crossing drawing is NP-complete [GT95], [SSV94]. By contrast, for the projective drawing versions of these problems we provide polynomial time solutions.

Lemma 2 *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.*

Proof: Given two line segments (edges of S) in E^3, all directions $\vec{d}$ 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. (Again refer to Figure 1.) This tetrahedron in turn determines four great-circles on the unit sphere of directions that define a pair of antipodal convex spherical quadrilaterals. Thus each pair of segments of S yields a pair of antipodal spherical quadrilaterals on the direction sphere that correspond to a set of directions which results in a crossing occurring between these two line segments. Such quadrilaterals are termed *forbidden*. Then a crossing-free direction of projection must correspond to a point which is outside of any forbidden quadrilateral. Conversely, a point belonging to any forbidden quadrilateral must give a projection with at least one crossing.

■

We proceed by solving the problem of deciding whether a set of line segments admits a crossing-free projection.

Theorem 5 *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 $O(n^2 + k)$ space, where k is the number of edge intersections of forbidden quadrilaterals and $k = O(n^4)$.*

Proof: The set of $O(n^2)$ forbidden spherical quadrilaterals given by all pairs of segments in E^3 determines a spherical arrangement on the sphere of directions. 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 $\vec{Op}$. Notice that on each face the intersection of the forbidden quadrilaterals is either a convex set or the empty set. Although this representation is not standard, it will allow us to use many algorithms which only work for straight-line polygons. To determine if S admits a crossing-free projection then reduces to the problem of determining if the transformed straight-line quadrilaterals cover the cube. We can do this by computing the *contour of the union* of these quadrilaterals. Note that the union may no be simply connected (i.e. it may contain holes) however, if the contour of the union is empty, then there is no direction that yields a projection without crossings (i.e. every point on the cube is covered by at least one quadrilateral).

Several algorithms have been developed for computing the contour of the union of a set of polygons. Some of these [SBS92], [CNL89], [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 edges in the polygons and k is the number of intersections of the edges. Souvaine & Bjorling-Sachs [SBS92] 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].

■

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

Besides the obvious application of minimum-crossing projections to visualization, we mention here that they also have applications to point location problems in 3-D. Consider a 3-D convex subdivision of space. Recall that the point location algorithm of Preparata & Tamassia [PT92] projects the skeleton of the subdivision onto the xy-plane to obtain a new planar subdivision with additional vertices at all intersection points. This planar subdivision is then pre-processed for planar point location before doing binary search on the z direction. We can apply our algorithm to the original subdivision to minimize the memory required by the planar point location portion of their algorithm.

4 Monotonic projections

The general notion of monotonicity is another characteristic of polygonal objects that aids in their visualization. In fact, projections that preserve monotonicity of trees finds application in medical imaging (see Sun et al. [SLG94]). Veins and arteries in the body are 3D tree-like structures that sometimes are monotonic, and preserving this monotonicity aids in 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 answering 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 orthographic projective versions of these drawing problems. First we consider the monotonicity of polygonal chains in E^3. Specifically, we address three questions. Given a polygonal chain P and a direction $\vec{d}$, is P monotonic with respect to direction $\vec{d}$? Recall that a polygonal chain $P = v_1, v_2, \cdots, v_n$ is *monotonic* in direction $\vec{d}$ provided that the intersection of P with every plane with normal $\vec{d}$ is empty, or a point. 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 with respect to which P is monotonic in $O(n \log n)$ time.

Given two points a and b, let $\vec{ab}$ denote the vector directed from a to b. A plane can be defined by a point p contained in that plane and the normal vector $\vec{n}$ of the plane. Given a point p and a vector $\vec{n}$, the plane defined by them is denoted by $H(p, \vec{n})$. Given a plane $h = H(p, \vec{n})$, we define the two half-spaces determined by this plane as follows. The open and closed half-spaces h^+ are defined as $\{x \mid \vec{px} \cdot \vec{n} > 0\}$ and $\{x \mid \vec{px} \cdot \vec{n} \geq 0\}$, respectively. Similarly, the open and closed half-spaces h^- are defined as $\{x \mid \vec{px} \cdot \vec{n} < 0\}$ and $\{x \mid \vec{px} \cdot \vec{n} \leq 0\}$, respectively. Henceforth, all half-spaces are open unless explicitly stated otherwise. To avoid ambiguity and simplify the discussion, we adopt the convention that if P is monotonic in direction $\vec{d}$, then v_1 is a minimum for P with respect to $\vec{d}$. 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 direction $\vec{d}$ is that their sub-chains are also monotonic with respect to $\vec{d}$.

The above implies that it suffices to determine all directions with respect to which a line segment is monotonic in order to compute all directions for which a polygonal chain is monotonic. Now a line segment is monotonic in every un-oriented direction except those perpendicular to the line segment. By our convention, we are interested in the oriented 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 $D = \{\vec{d} \mid \vec{d} \cdot \vec{ab} > 0\}$. Let $h = H(O, \vec{ab})$ (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 S^2 that represents all directions in space (the sphere of directions). Given a polygonal chain $P = v_1, v_2, \cdots, v_n$ and a direction $\vec{d}$, we would like to determine if P is monotonic with respect to $\vec{d}$. We simply verify that each of the line segments of $P = [v_1, v_2], [v_2, v_3] \cdots, [v_{n-1}, v_n]$ is monotonic with respect to $\vec{d}$, i.e., that $\vec{v_j v_{j+1}} \cdot \vec{d} > 0$. We conclude with the following.

Theorem 7 *Given a polygonal chain P and a direction $\vec{d}$, in $O(n)$ time, one can determine if P is monotonic with respect to $\vec{d}$.*

We can determine if a polygonal chain $P = v_1, v_2, \cdots, v_n$ is monotonic with respect to some direction in the following way. Let h_i^+ represent the half-space determined by the plane $H(O, \vec{v_i v_{i+1}})$. Let D be the intersection of the h_i^+ over all i. Then the set of all directions for which P is monotonic is described by $D \cap S^2$. Determining if D, the intersection of a set of half-spaces is non-empty can be accomplished in linear time using linear programming [Meg83]. If the intersection of the halfspaces is nonempty, then it also intersects the sphere of directions. Therefore we conclude with the following.

Theorem 8 *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 [PS85], we conclude with the following.

Theorem 9 *Given a polygonal chain P, one can determine in $O(n \log n)$ time all the directions with respect to which P is monotonic.*

Now we turn to 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]. 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 $EA(v)$.

Observation 1 *Vertex v is a minimum with respect to $\vec{d}$ for $EA(v)$ if and only if $\forall e \in EA(v)$, v is a minimum with respect to $\vec{d}$ for e.*

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 $\vec{e}$ we denote the vector $v\vec{v}_i$. Let $h(e) = H(O, \vec{e})$. We see that $MD(v)$ is the intersection of $h(e)^+$ over all e contained in $EA(v)$. A vertex v of a geometric graph is a *proper local minimum* with respect to direction $\vec{d}$ provided that v is a minimum for the set $EA(v)$ in direction $\vec{d}$. A vertex v is a *local minimum* with respect to direction $\vec{d}$ if $\forall e \in EA(v)$, the edge e is contained in the closure of $H^+(v, \vec{d})$.

We now address several questions concerning the monotonicity of trees. Suppose we are given a rooted tree T, and a direction $\vec{d}$. The first question we address is to determine if T is monotonic in direction $\vec{d}$. 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 $\vec{d}$ with respect to which T is monotonic? (2) Given an unrooted tree T and direction $\vec{d}$, does there exist a vertex $v \in T$ such that T rooted at v is monotonic with respect to $\vec{d}$? (3) Given an unrooted tree T, does there exist a direction $\vec{d}$ and a vertex $v \in T$ such that T rooted at

v is monotonic with respect $\vec{d}$? (4) Given an unrooted tree T, find all vertices $v \in T$ and directions $\vec{d}$ such that T rooted at v is monotonic in direction $\vec{d}$.

Recall that a tree T is called a *rooted tree* if a unique vertex v of T is specified as its root, otherwise the tree is *unrooted* or *free*. A rooted tree T is monotonic in direction $\vec{d}$ provided that the path from the root to every vertex is monotonic in direction $\vec{d}$. The key behind the efficient solution of all above-mentioned problems depends on the following characterization of the monotonicity of rooted trees.

Lemma 3 *A rooted tree T is monotonic in direction $\vec{d}$ if and only if the root r of T is a proper local minimum and no other vertex is a local minimum with respect to direction $\vec{d}$.*

Proof:

($\Longrightarrow$) Assume T is monotonic with respect to $\vec{d}$. If r is not a proper local minimum then at least one root to leaf path in T is not monotonic. Suppose there exists a vertex v of T that is not the root, such that v is a local minimum. Let $h = H(v, \vec{d})$. We see that $EA(v)$ must be in the closure of h^+ since v is a local minimum with respect to $\vec{d}$. 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 $\vec{d}$, by the convention of monotonic paths. Therefore, $P - \{v\}$ must be contained in h^-. Let v_j be the vertex preceding v in P. Since v_j is adjacent to v, we see that $[vv_j] \in EA(v)$. But this implies that $[vv_j]$ is contained in the closure of h^+ which contradicts the monotonicity of P.

($\Longleftarrow$) Given a vertex, we will represent the plane $h = H(v, \vec{d})$ by h_v. Assume that the root r of T is the only proper local minimum and no other vertex of T is a local minimum. Let v be an arbitrary vertex of T. We will show that the path P from r to must be monotonic in direction $\vec{d}$. Let the path $P = \{v_1(= r), v_2, \cdots, v_{k-1}, v_k(= v)\}$. Suppose P is not monotonic with respect to $\vec{d}$. Let v_i $(1 < i \leq k)$ be the first vertex of P such that $[v_{i-1}v_i]$ is not monotonic with respect to $\vec{d}$. Then v_{i-1} must be contained in the closure of $h^+_{v_i}$. Since v_i is not a local minimum, and T is acyclic, there must exist a vertex v_j in $EA(v_i)$ different from v_{i-1}, 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 $EA(v_j)$ different from v_j, 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 the closure of $h^+_{v_t}$, contradicting the fact that no vertex of T other than the root is a local minimum.

■

Suppose we are given a tree T, a root r of T and a direction $\vec{d}$, and want to determine whether T is monotonic with respect to $\vec{d}$. By Lemma 3, since r is the only proper local minimum and no other vertex is a local minimum with respect to $\vec{d}$, the direction $\vec{d}$ cannot be contained in the closure of $MD(v)$ for all vertices of T other than the root. Since $MD(v)$

is an intersection of half-spaces, determining whether or not a direction $\vec{d}$ is contained in $MD(v)$ can be done in $O(|MD(v)|)$ time. But $O(|MD(v)|)$ 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 $\vec{d}$.

Theorem 10 *Given a rooted tree T, and a direction $\vec{d}$, one can decide in $O(n)$ time if T is monotonic with respect to $\vec{d}$.*

Suppose next that the root of the tree is no longer specified. Then, to determine if T is monotonic in direction $\vec{d}$, we must first find a root. By Lemma 3, the root must be the only proper local minimum with respect to direction $\vec{d}$ and no other vertex is a local minimum.

Therefore, there must exist exactly one vertex r of T such that $\vec{d}$ is contained in $MD(r)$, which becomes the root. All other vertices v of T must have the property that $\vec{d}$ is not in the closure of $MD(v)$. Again this can be determined in linear time. Therefore, we conclude with the following.

Theorem 11 *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 $\vec{d}$.*

As before, we use the cube AC for representing directions of projection. Now, suppose that the root r of T is specified but the direction is not. For T to be monotonic in some direction $\vec{d}$, we see that $\vec{d}$ 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, one needs to 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 proper 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 $I_i = \{I(i,j) \mid 1 \leq j \leq n\}$ 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 the interior of k polygons of the set I_i, then there are k vertices of T that are local minima with respect to the direction $\vec{Op}$ and each of the vertices that are proper local minima is identified by the polygon which contains $\vec{Op}$. That is, if p is contained in polygon $I(i,3)$ then vertex v_3 is a proper local minimum with respect to direction $\vec{Op}$. 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 $F_i(1 \leq i \leq 6)$ that are covered by only one polygon of the set $I_i(1 \leq i \leq 6)$. 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 $O(n \log n + k)$ time where k 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 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 is known as the planar dual graph of A_i (see [BM76] for more information on duals of planar graphs). 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_l, a_m) we traverse corresponds to going inside or outside a polygon of I_i, in which case we take the number of a_l 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 12 *In $O(n^2)$ time, we can determine all directions and roots with respect to 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 $\vec{d}$ provided there exist two vertices u, v of P such that both paths from u to v are monotonic in direction $\vec{d}$. 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.

Corollary 1 *Given a simple polygon P in E^3 and a direction $\vec{d}$ in $O(n)$ time it can be determined if P is monotonic with respect to $\vec{d}$.*

Corollary 2 *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

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. Our results also have application to graph drawing for knot-theorists. Let K be a knot with n vertices. To study the knot combinatorial properties, knot theorists obtain a planar graph G called the diagram of K by a regular projection of K. Many of their algorithms are applied to G and therefore their time complexity depends on the space complexity of G. By combining our algorithms we can obtain regular projections with the minimum number of crossings, thereby minimizing the time complexity of their algorithms.

6 Acknowledgments

The authors thank Suneeta Ramaswami, Fady Habra, Ivan Balaban, Bernard Chazelle and Mike Goodrich for correspondence. We also thank Ferran Hurtado for useful discussions. We thank David MacDonald and Rick Hoge of the Montreal Neurological Institute and McGill University for providing data and images. The second author thanks Lola for her support.

References

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

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

[Bal95] I. Balaban. An optimal algorithm for finding segments intersections. *Proc. Eleventh Annual ACM Symp. on Computational Geometry*, 211-219, 1995.

[BDE96] G. Barequet, M. Dickerson and D. Eppstein. On Triangulating Three-Dimensional Polygons. *Proc. 12th Annual ACM Sympos. Comp. Geom.*, 38–47, 1996.

[BGK95] T. Burger, P. Gritzman, and V. Klee. Polytope projections. *TR No 95-14, Dept. Mathematics, Trier University*, 1995.

[BM76] J.A. Bondy and U.S.R. Murty. *Graph Theory and Applications.* North-Holland, New York, 1976.

[BO79] J.L. Bentley, and T.A. Ottman. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, 8:300-307, 1979.

[Bor69] U. Borsuk. *Multidimensional Analytic Geometry.* Polish Scientific Publishers, 1969.

[BR94] P. Bhattacharya, and A. Rosenfeld. Polygons in three dimensions. *Journal of Visual Communication and Image Representation*, 5(2):139-147, 1994.

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

[CNL89] K. Chiang, S. Nahar, and C. Lo. Time-efficient VLSI artwork analysis algorithms in Goalie2. *IEEE Trans. Computer-Aided Design*, 39:640-647, 1989.

[Col90] C. Colin. Authomatic computation of a scene's good views. *MICAD '90*, 1990.

[DBETT94] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for graph drawing: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4:235-282, 1994.

[FvDFH90] J.D. Foley, van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 1990.

[Gal95] R.S. Gallagher, editor. *Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis.* IEEE Computer society Press, 1995.

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

[GT95] A. Garg, and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *Proc. Graph Drawing '94, Lecture Notes in Computer Science*, 894:286-297, 1995.

[H92] J. Harris. *Algebraic Geometry* Springer-Verlag, 1992.

[HCV83] Hilber, and Cohn-Vossen. *Geometry and Imagination.* Chelsea Publishing Company, 1983.

[HMTT94] T. Hirata, J. Matousek, X.-H. Tan, and T. Tokuyama. Complexity of projected images of convex subdivisions. *Computational Geometry*, 4:293-308, 1994.

[Kel93] M.M. Keller, and P.R. Keller. *Visual Cues: Practical Data Visualization.* IEEE Computer Society Press, 1993.

[KK88] T. Kamada, and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics and Image Processing*, 41:43-56, 1988.

[Liv93] Livingston. *Knot Theory.* Vol. 4 of *The Carus Mathematical Monographs.* The Mathematical Association of America, 1993.

[Meg83] N. Megiddo. Linear-time algorithms for linear programming in R^3. *SIAM Journal of Computing*, 12:759-776, 1983.

[MS85] M. McKenna, and R. Seidel. Finding the optimal shadows of a convex polytope. *Proc. ACM Symposium on Computational Geometry*, 24-28, 1985.

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

[PS85] F. Preparata and M.I. Shamos. *Computational Geometry.* Springer-Verlag, 1985

[PT92] F. Preparata, and R. Tamassia. Efficient point location in a convex spatial cell-complex. *SIAM Journal of Computing*, 21:267-280, 1992.

[Rei32] R. Reidemeister. *Knotentheorie.* Ergebnisse der Mathematic. Springer-Verlag, 1932. L.F. Boron, C.O. Christenson and B.A. Smith (English translation) *Knot Theory*, BSC Associates, Moscow, Idaho, 1983.

[SBS92] D. Souvaine, and Bjorling-Sachs. The contour problem for restricted-oriented polygons. *Proc. of the IEEE*, 80:1449-1470, 1992.

[SSV94] F. Shaharokhi, L. Szekely, and i. Vrt'o. Crossing numbers of graphs, lower bounds techniques and algorithms: a survey. *Lecture Notes in Computer Science*, 894:131-142, 1994.

[Str82] G. Strang. The width of a chair. *The American mathematical Monthly*, 89(8):529-534, 1982.

[SLG94] Y. Sun and I. Liu and J.K. Grady. Reconstruction of 3D tree-like structures from three mutually orthogonal projections. *IEEE Trans. Pat. Ana. Mach. Intell.*, 16: 241-248, 1994.

[Tou85] G. Toussaint. *Movable separability of sets.* Computational Geometry North Holland, 1985.