Theoretical Computer Science 35 (1985) 55–69
North-Holland

FINDING TRANSVERSALS FOR SETS OF SIMPLE GEOMETRIC FIGURES

Herbert EDELSBRUNNER
Institute for Information Processing, Technical University of Graz, A-8010 Graz, Austria

Communicated by M.S. Paterson
Received August 1982
Revised May 1984

Abstract. A straight line that intersects all members of a set S of objects in the real plane is called a transversal of S. Geometric transforms are described that reduce transversal problems for various types of objects to convex hull problems for points. These reductions lead to efficient algorithms for finding transversals which are also described. Applications of the algorithms are found in computer graphics: "Reproduce the line displayed by a collection of pixels", and in statistics: "Find the line that minimizes the maximum distance from a collection of (weighted) points in the plane".

1. Introduction

Let S denote a set of objects in the d-dimensional Euclidean space E^d, for some positive integer d. A $((d-1)$-dimensional) hyperplane in E^d is a *transversal of S* if it intersects all objects of S. Classical Helly-type theorems of the following generic form imply trivial algorithms for deciding the existence of transversals [9, 2]): For S a finite set of objects with certain properties, there exists a transversal if any k objects of S admit a transversal. Typically, k is a rather small constant so that polynomial time algorithms follow.

Departing from this mathematically beautiful but computationally expensive characterization, Edelsbrunner, Overmars and Wood [5] develop a method for planar visibility problems that yields a rather general method for computing transversals in E^2 in $O(n^2 \log n)$ time, for $n = |S|$. $O(n \log n)$ time is shown to suffice for the special cases of vertical line segments [12] and also for line segments with arbitrary directions [4].

This paper uses geometric transforms to cast transversal problems into better understood convex hull problems. Section 2 presents a collection of preliminary results needed in Sections 3 to 5. The insight gained by this novel approach leads to efficient algorithms for finding transversals for families of objects in E^2 and higher dimensions. Section 3 considers axis-parallel hyper-rectangles in E^d and applies methods from linear programming to find a transversal in $O(n)$ time. An application of the method to computer graphics is presented. Then, Section 4 considers translates of a simple object, in a sense to be made precise, in E^2. An application of the methods to a problem in statistics is demonstrated. Section 5 focusses on homothets of a simple object in E^2, that is, on objects that derive from an original object by translation and changing the size. Finally, the results and methods are discussed in Section 6 which also singles out a few open problems.

2. Preliminaries

We find it convenient to briefly discuss preliminary algorithms for constructing convex hulls of sets of points and for computing separating hyperplanes. The *convex hull* of a set of points in E^d is the smallest convex polytope that contains all points of the set. Efficient computational solutions for constructing the convex hull of a finite set of points are known, which imply, by the transforms and methods to be described, efficient solutions for transversal problems. We state the relevant results.

Proposition 2.1. *The convex hull of a set of n points in* E^2 *can be constructed in* $O(n \log n)$ *time and* $O(n)$ *space.*

Algorithms that verify the assertion are given in [7, 13, 14], etc. Only the method of Preparata and Hong [13] generalizes to E^3 without losing efficiency.

Proposition 2.2. *The convex hull of n points in* E^3 *can be constructed in* $O(n \log n)$ *time and* $O(n)$ *space. Additional* $O(n)$ *time of preprocessing suffices to allow the computation of a tangent plane with given normal vector in* $O(\log n)$ *time.*

The method for computing tangent planes with given direction is described in [3].

A hyperplane *separates* two sets S and T of points in E^d if it intersects every open line segment connecting a point of S with a point of T. The linear programming methods in [11] imply the following.

Proposition 2.3. $O(n)$ *time suffices to find a separating hyperplane of two sets of a total of n points in* E^d *(if it exists).*

3. Transversals for rectangles

We call an object in E^d a *(d-dimensional) rectangle* if it is the Cartesian product of d open intervals, one on each coordinate axis. A rectangle is completely determined by its *lower corner* (the Cartesian product of the startpoints of the defining intervals) and its *upper corner* (determined by the endpoints of the intervals).

The following generalization of a result in [9] implies a trivial algorithm that decides the existence of a transversal for a set S of n rectangles in E^d in $O(n^{d+1})$ time:

We call a hyperplane in E^d specified by $x_d = a_1x_1 + \cdots + a_{d-1}x_{d-1} + a_d$ *negative* if the real numbers a_i, for $i = 1, \ldots, d-1$, are nonpositive (see Fig. 3.1). There is a negative transversal for S if there is a negative transversal for any $d+1$ rectangles of S.

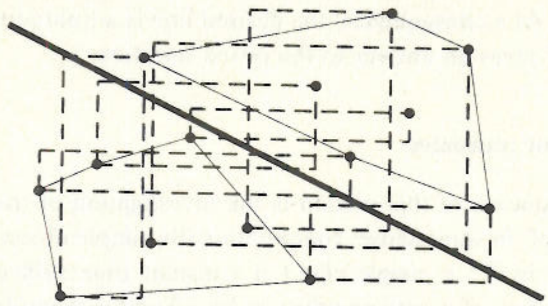

Fig. 3.1. Negative transversal for seven rectangles.

This algorithm can be improved considerably as is shown below. An additional advantage of the method to be described is the possibility to actually compute a transversal (if it exists).

Theorem 3.1. $O(n)$ *time suffices to find a transversal of a set* S *of* n *rectangles in* E^d (*if it exists*).

Proof. We reduce the problem to finding a separating hyperplane for sets of points in E^d. Without loss of generality, only negative transversals are considered. A negative hyperplane h intersects a rectangle r if and only if the lower corner of r is below h and the upper corner is above h. Let L and U be the sets of lower and upper corners of rectangles in S, respectively. Then a negative hyperplane h is a transversal of S if and only if h separates L from U (see Fig. 3.1). The assertion follows from Proposition 2.3. □

We note that the described method extends to transversal problems for sets of so-called k-oriented objects [8]: Let k be some positive constant integer and let K be a collection of k distinct directions. A convex polytope in E^d is *k-oriented* (w.r.t. K) if the direction of the normal of each facet is in K.

We close this section with an application to a problem in computer graphics. The following algorithm is frequently used to display a straight line L on a raster display device [6]:

> Out of each column of points (pixels) choose the one closest to L to represent a point of L.

Now assume that a set S of points is given. How fast can we decide whether or not S displays a line, and, if the answer is affirmative, how fast can such a line be determined? To answer these questions, we note that a point p on the screen represents a point of a line L if and only if p is closer to L than are the points a and b immediately above and below p. Thus, L intersects the vertical segment connecting $\frac{1}{2}(p+a)$ and $\frac{1}{2}(p+b)$. Each point of S defines a vertical segment which must intersect L. As a consequence, the desired line is a transversal of the segments and Theorem 3.1 gives an answer to the posed question.

4. Transversals for translates

The primary concern of this section is the investigation of transversal problems for uniform sets of, in some sense, computationally simple objects. We call an open convex subset o in E^2 a *simple object* if constant time suffices to compute the orthogonal projection of o onto an arbitrary line. Typical examples are open convex polygons with a constant number of edges, open discs, open ellipses, etc. Using the natural extension of sums of points (or vectors) to sums of sets of points, a planar object t is called a *translate* of another object o it there is a vector v such that $t = o + v$ (see Fig. 4.1).

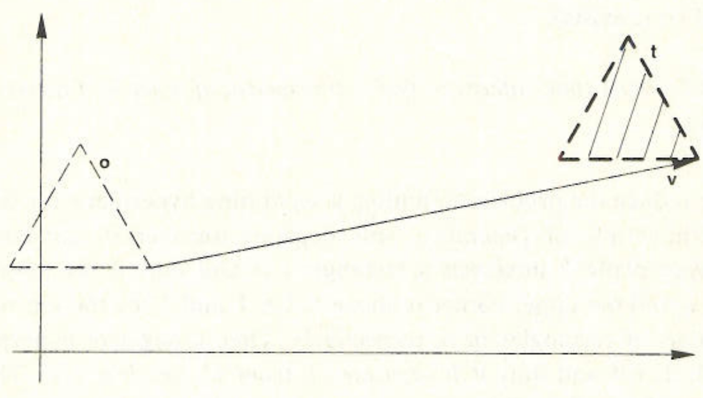

Fig. 4.1. Object o with translate t.

We are interested in solving transversal problems for finite sets of translates of some given simple object o. Note that a set S of translates of o is uniquely defined by o and the vectors that translate o. We call o the *prototype of S*. The vector v that defines $t = o + v$ also identifies the unique point $P(t) = O + v$, with O the origin of E^2. We call $P(S) = \{P(t) \mid t \text{ in } S\}$ the (*corresponding*) *point-set of S* (see Fig. 4.2).

For M an arbitrary line, we write A^M for the orthogonal projection of a set A in E^2 onto M. Then (o^M, O^M) is termed the *basic projection of M* and, by definition, can be computed in constant time. A pair (i, p) with i an interval and p a point on M, is termed a *mirror image* of the basic projection if there is a point m on M such that $o^M - m = m - i$ and $O^M - m = m - p$ (see Fig. 4.3). Let now L denote an arbitrary line, M some line perpendicular to L, and p_0 the intersection of L and M. The *stripe* $\mathrm{ST}(L)$ *of L* is the set of points q such that i_0 contains q^M, with (i_0, p_0) a mirror image of the basic projection (see Fig. 4.3).

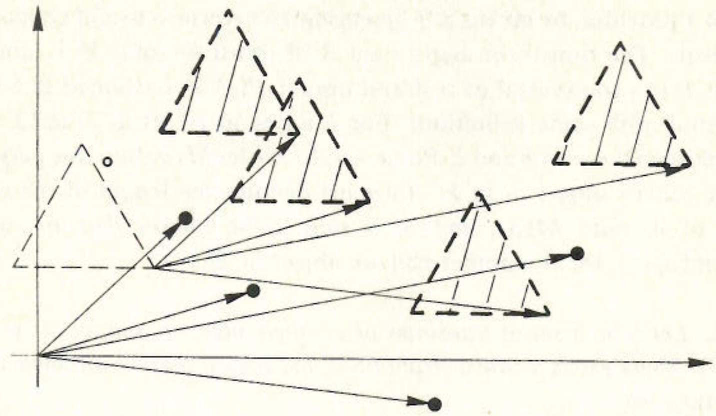

Fig. 4.2. Set of translates and corresponding point-set.

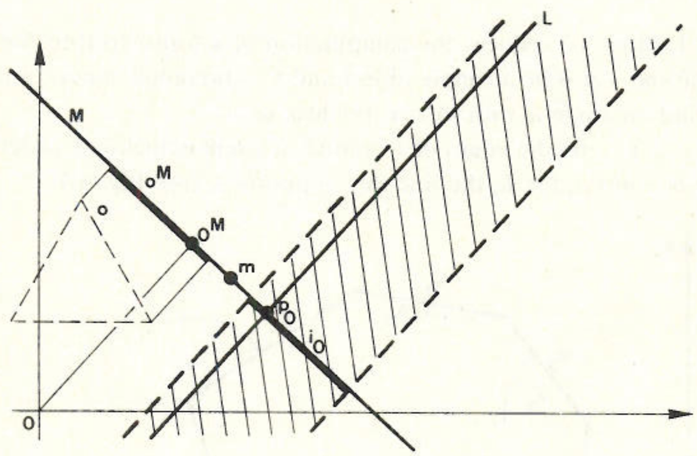

Fig. 4.3. Basic projection of M and stripe of L.

Since the basic projection of M can be computed in constant time, the same is true for $\mathrm{ST}(L)$. The following lemma explicates the relation between the corresponding point-set of a set of translates and the stripe of a line.

Lemma 4.1. *Let t be a translate of some prototype. A line L intersects t if and only if $P(t)$ lies in* $\mathrm{ST}(L)$.

Proof. Let M be some line perpendicular to L. L intersects t if and only if p_0 $(=L^M)$ is contained in t^M. Now, p_0 is in t^M if and only if $P(t)^M$ is in i_0, with (i_0, p_0) a mirror image of $(t^M, P(t)^M)$ and therefore $i_0 = \mathrm{ST}(L)^M$. □

Lemma 4.1 provides, by means of a geometric transform, a useful characterization of transversals. The transform maps a set S of translates into $P(S)$ and a line L into $\mathrm{ST}(L)$. L is a transversal of S if and only if $P(S)$ is contained in $\mathrm{ST}(L)$.

We continue with some definitions: For L a line in E^2 let $a(L)$ in $[0, \pi)$ denote the angle between the x-axis and L. For $a = a(L)$, we let $M(a)$ be a line perpendicular to L. For a convex object o_1 in E^2, $\mathrm{th}(a, o_1)$ denotes the length of the orthogonal projection of o_1 onto $M(a)$, and o_1 is said to be *thicker than* o_2 (w.r.t. a) if $\mathrm{th}(a, o_1) > \mathrm{th}(a, o_2)$, for o_2 another convex object in E^2.

Lemma 4.2. *Let S be a set of translates of a simple object o and let C be the convex hull of $P(S)$. There exists a transversal of S if and only if there is an angle a such that* $\mathrm{th}(a, C) < \mathrm{th}(a, o)$.

Proof. If L is a transversal of S, then C is contained in $\mathrm{ST}(L)$. Since $\mathrm{th}(a(L), \mathrm{ST}(L)) = \mathrm{th}(a(L), o)$, we conclude that $\mathrm{th}(C) < \mathrm{th}(o)$. Conversely, an angle b with $\mathrm{th}(b, C) \geq \mathrm{th}(b, o)$ does not permit a transversal perpendicular to $M(b)$. □

Note that Lemma 4.2 reduces the computation of a transversal to the following *thickness problem*: Let o be a simple object and C a bounded convex polygon with n vertices; find an angle a with $\mathrm{th}(a, C) < \mathrm{th}(a, o)$.

Let $e_0, e_1, \ldots, e_{n-1}$ be the edges of C sorted in counterclockwise order. For each e_i we define $a_i = a(L_i)$, for L_i the line that supports e_i (see Fig. 4.4).

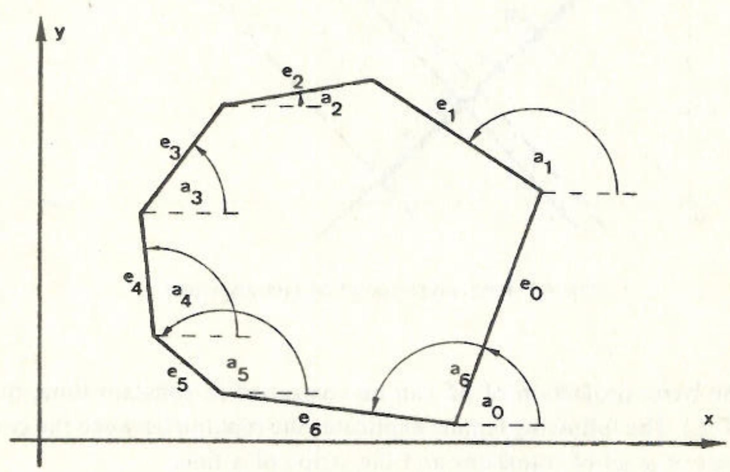

Fig. 4.4. Polygon with associated angles.

Lemma 4.3. *If there exists an angle a with* $\mathrm{th}(a, C) < \mathrm{th}(a, o)$, *then there is an index* k, $0 \leq k \leq n-1$, *such that* $\mathrm{th}(a_k, C) < \mathrm{th}(a_k, o)$.

Proof. Rename the angles associated with the edges of C as $b_0, b_1, \ldots, b_{n-1}$ such that $b_i \leq b_{i+1}$, for $i = 0, \ldots, n-2$. We assume $\mathrm{th}(a, C) < \mathrm{th}(a, o)$ for some angle a, and $\text{th}(b_i, C) \geq \text{th}(b_i, o)$ for all $0 \leq i \leq n-1$. Without loss of generality let $b_0 < a < b_1$. Since C is at least as thick as o w.r.t. b_0 and b_1, there is a translate of o contained in the quadrangle Q defined by the four supporting lines of C whose angles are b_0 and b_1 (see Fig. 4.5).

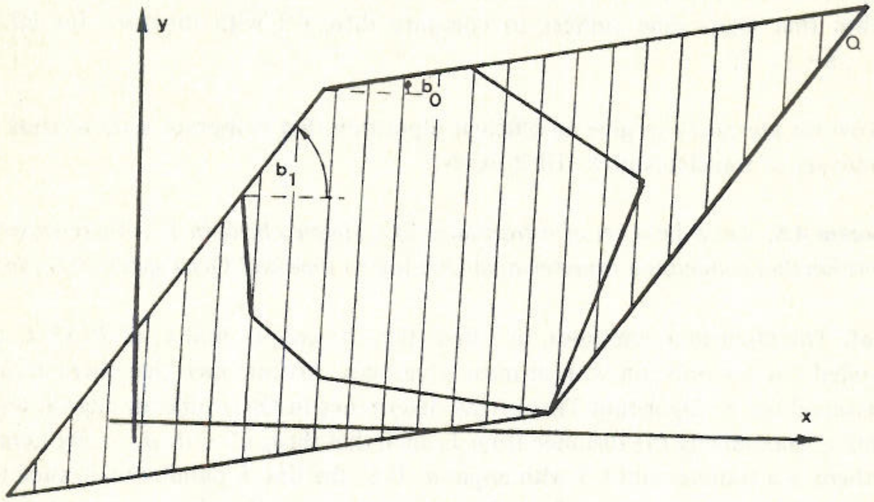

Fig. 4.5. Quadrangle defined by b_0 and b_1.

Since there is no angle between b_0 and b_1 which is associated with an edge of C, $\text{th}(b, Q) = \text{th}(b, C)$ for all $b_0 < b < b_1$. This contradicts the assumption that $\text{th}(a, C) < \text{th}(a, o)$, for some a with $b_0 < a < b_1$. □

Lemma 4.3 allows the design of an efficient algorithm for the thickness problem.

Theorem 4.4. *Let C be a convex polygon with n vertices and let o be a simple object. There exists an algorithm that finds an angle a with* $\text{th}(a, C) < \text{th}(a, o)$ *in* $O(n)$ *time (if it exists).*

Proof. By Lemma 4.3, the thickness problem reduces to identifying an angle a associated with an edge of C such that $\text{th}(a, C) < \text{th}(a, o)$. An algorithm that is similar to the one designed in [14] for determining the diameter of a convex polygon determines $\text{th}(a, C)$ for each $a = a_i$ in $O(n)$ time:

Algorithm Thickness. Let $v_0, \ldots, v_{n-1}$ denote the vertices of C such that v_i and v_{i+1} are the endpoints of edge e_i (taking all indices modulo n).

Step 1. Determine vertex v_k that maximizes the distance from line L_0 supporting e_0. This distance equals $\text{th}(a_0, C)$. Set $i = 0$.

Step 2. Let e_i and v_j denote respectively the edge and the vertex such that $\text{th}(a_i, C)$ is the distance of v_j from L_i. Determine the smallest positive integer m such that the distance of v_{j+m} to L_{i+1} is smaller than the one of v_{j+m-1}. By convexity of C, $\text{th}(a_{i+1}, C)$ is the distance of v_{j+m-1} from L_{i+1}. If $i+1$ is still smaller than $n-1$, then repeat Step 2 with $i=i+1$.

By definition, $\text{th}(a_i, o)$ can be determined in constant time for $0 \leq i \leq n-1$. This implies that $O(n)$ time suffices to compare $\text{th}(a_i, C)$ with $\text{th}(a_i, o)$, for all $i = 0, \ldots, n-1$. □

Now we are ready to give an efficient algorithm that computes a transversal for a finite set of translates in E^2 (if it exists).

Theorem 4.5. *Let S be a set of n translates of a simple object in E^2. There exists an algorithm that computes a transversal in* $O(n \log n)$ *time and* $O(n)$ *space (if it exists).*

Proof. The algorithm computes, in a first step, the convex hull C of $P(S)$. C is a bounded convex polygon with at most n vertices. Assume now that there exists a transversal for S. Algorithm THICKNESS determines in $O(n)$ time an edge e_i and a vertex v_j maximizing the distance from L_i such that $\text{th}(a_i, C) < \text{th}(a_i, o)$. By Lemma 4.3 there is a transversal of S with angle a_i. E.g., the line L parallel to L_i such that v_i, v_{i+1}, and v_j are equidistant from L is a transversal of S and can be computed in constant time from e_i and v_j. □

It seems worthwhile to note that the described method does not generalize to E^3 since Lemma 4.3 does not. We define the *thickness* of a convex polytope P in E^3 as the length of the shortest orthogonal projection of P onto a line in E^3. Then $2^{-1/2}$ is the thickness of the regular tetrahedron with edges of length 1. However, each face has a vertex $(2/3)^{1/2} > 2^{-1/2}$ units of length away. Thus, the thickness of P cannot be computed by checking only the directions determined by faces of P.

There is an interesting application of Algorithm THICKNESS to a problem in statistics aimed at computing regression lines. It is trivial to modify Algorithm THICKNESS such that it computes the *breadth* $B(C)$ of a convex polygon C, that is, $B(C) = \min\{\text{th}(a, C) \mid a \text{ in } [0, \pi)\}$. (Notice that Lemma 4.3 implies that if $B(C) = \text{th}(a, C)$, then a is associated with an edge of C.) Let L_C be a line with angle $a = a(L_C)$ such that $B(C) = \text{th}(a, C)$ and that L_C pierces C in the middle, that is, L_C^M is the center of C^M, for M perpendicular to L_C. If C contains n vertices, then L_C can be computed in $O(n)$ time from C (trivial extension of Theorem 4.4).

Let now S be a set of n points in E^2 with the hypothesis that the points represent observations of some affine dependence of y on x. For a line L, let $\text{dev}(L) = \max\{d(p, L) \mid p \text{ is } S \text{ and } d(p, L) \text{ denoting the orthogonal distance of } p \text{ from } L\}$ be called the *deviation* of L. For C the convex hull of S, L_C minimizes the deviation and thus approximates S best in the minmax sense. We suspect that the efficiency of the sketched algorithm makes it an interesting alternative to existing methods for computing regression lines in E^2 [10].

5. Transversals for homothets

An object h in E^2 is called a (*positive*) *homothet* of another object o if there exists a vector v and a positive real number m such that $h = mo + v$ (see Fig. 5.1).

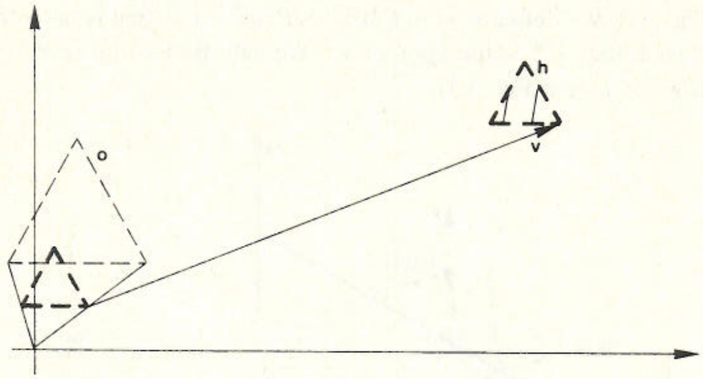

Fig. 5.1. Object o with homothet h in E^2.

This section concentrates on transversal problems for sets of homothets of a simple object. The methods remain the same as those used for translates: A transform is exploited to reduce transversal problems to convex hull problems. The additional degree of generality, expressed by the factor of magnitude m, will be reflected by an additional dimension of the obtained convex hull problems.

Let S be a set of homothets of a simple object o in E^2. We call o the *prototype* of S. For convenience, E^2 is identified with the xy-plane in E^3. For a homothet $h = mo + v$, with $v = (v_1, v_2)$, we call $P(h) = (v_1, v_2, m)$ the (*corresponding*) *point* of h, and $P(S) = \{P(h) | h \text{ in } S\}$ the (*corresponding*) *point-set* of S. The unbounded cone $C(h)$ with apex $P(h)$ and h the intersection of $C(h)$ with the xy-plane is termed the *cone* of h (see Fig. 5.2). Recall that $C(h)$ is open since h is open.

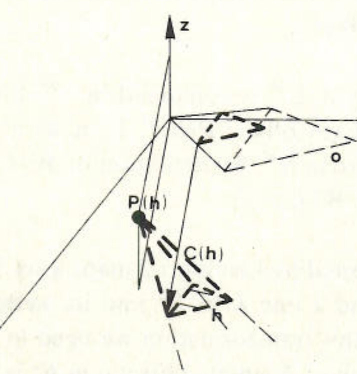

Fig. 5.2. Homothet, point, and cone.

Observation 5.1. *Let h_1 and h_2 be two homothets of a common object. Then $C(h_1)$ and $C(h_2)$ are translates of each other.*

Let now L be some line in the xy-plane and let M be a plane perpendicular to L. The orthogonal projection $C(o)^M$ of $C(o)$ onto M is called the *basic projection* of M (see Fig. 5.3). We define $w' = -(C(o)^M - P(o)) + L^M$, that is, w' is the translate of $-C(o)^M$ such that L^M is the apex of w'. We call $W(L) = \{p \text{ in } E^3 \mid w' \text{ contains } p^M\}$ the *wedge* of L (see Fig. 5.3).

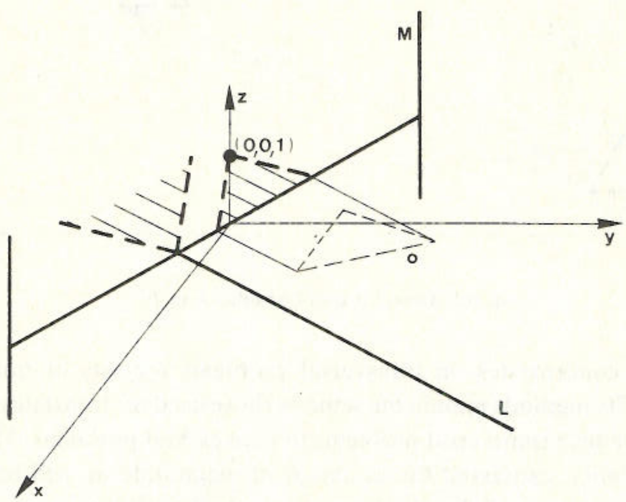

Fig. 5.3. Basic projection and wedge of L.

By definition of simple object, the basic projection of M and also $W(L)$ can be determined in constant time. The following assertion is a generalisation of Lemma 4.1 and describes the relation between the corresponding point-set of a set of homothets and the wedge of a line.

Lemma 5.2. *Let h be a homothet of a prototype in E^2. A line L in E^2 intersects h if and only if $P(h)$ is contained in $W(L)$.*

Proof. L intersects h if and only if L^M is contained in h^M. Since w' $(= W(L)^M)$ and $-C(h)^M$ are translates of each other, $P(h)^M$ is in w' if and only if L^M is contained in $C(h)^M$ and therefore in h^M. By definition of $W(L)$, $P(h)^M$ is in w' if and only if $P(h)$ is contained in $W(L)$. $\square$

The geometric transform suggested by Lemma 5.2 maps a set S of homothets into the set $P(S)$ of points in E^3, and a line L in E^2 into its wedge $W(L)$ in E^3. To develop an algorithm based on this transformation we need to be able to perform a particular primitive operation on o: A simple object o in E^2 is *tangible* if constant time suffices to compute the tangents on o that contain some arbitrary given point in E^2. We assume that no tangent exists if o contains the point, and that only one tangent is computed if the point lies on the boundary of o. Thus, the primitive operation can also be used to test whether or not o or its closure contains some given point.

The variant of the thickness problem to be solved for finding transversals reads as follows: Let P be a convex polytope above the xy-plane, and let o be a simple tangible object in the xy-plane; find a line L in the xy-plane such that $W(L)$ contains P. In order to attack this problem successfully, a geometric fact similar to Lemma 4.3 turns out to be useful. Some additional notation will simplify its discussion. By an *angle* we always mean an angle in the xy-plane as defined in Section 4. Let e be an edge of P connecting vertices $v = P(h_1)$ and $w = P(h_2)$. We call e *relevant* if neither $C(h_1)$ contains $C(h_2)$ nor the other way round, and there exists a line L in the xy-plane and a translate of $W(L)$ that contains the interior of P and e lies in a bounding halfplane of $W(L)$. The angle $a(L)$ defined by L is then termed an *angle of e*.

Lemma 5.3. *Each relevant edge has either one or two angles.*

Proof. Let p and q be the endpoints of a relevant edge e of polytope P, and let a be an angle of e. Then there is a line L with $a = a(L)$ such that a bounding halfplane of $W(L)$ contains p and q. This halfplane is tangent to both $C(h_1)$ and $C(h_2)$, for $P(h_1) = p$ and $P(h_2) = q$. Consequently, L is tangent to h_1 and h_2 and both homothets lie on the same side of L. Since h_1 and h_2 allow two common tangents of this kind, there are at most two angles of e. □

We now present the anticipated geometric fact.

Lemma 5.4. *Let o be a simple object in the xy-plane and let P denote a convex polytope above the xy-plane. If there is an angle a and a line L^* with $a = a(L^*)$ and P in $W(L^*)$, then there exists such an angle b of an edge of P.*

Proof. Let $b_0, b_1, \ldots, b_{k-1}$ be the angles of P's edges such that $b_i \leqslant b_{i+1}$, for $0 \leqslant i \leqslant k-2$. Without loss of generality, we assume the existence of a line L^* with $b_0 < a(L^*) < b_1$ such that $W(L^*)$ contains P. Furthermore, we assume that P is not contained in $W(L)$ for any line L with $a(L) = b_i$, for $0 \leqslant i \leqslant k-1$ and in particular for $i = 0, 1$.

For every angle b we define the wedge $W(b)$ as the intersection of all translates of $W(L)$, for L an arbitrary line with $a(L) = b$, that contain the interior of P. In addition, we let $\mathrm{ST}(b)$ be the intersection of $W(b)$ with the xy-plane. By assumption, $\mathrm{ST}(b_0)$ and $\mathrm{ST}(b_1)$ are nonempty and their intersection is a nonempty open quadrangle Q. Let r_0 and s_0 be the antipodal vertices of Q that allow tangent lines on Q with any angle in $[b_0, b_1]$. Let r and s be the two rays in the intersection of the boundaries of $W(b_0)$ and $W(b_1)$ that intersect the xy-plane in r_0 and s_0, respectively.

We argue below that there are vertices v and w of P on r and s, respectively, and that every wedge $W(a)$, with $b_0 < a < b_1$, which has v and w in the boundary intersects the xy-plane properly. Note that this contradicts the existence of L^* as assumed in the beginning and thus proves the assertion.

Fact 1. *Each of r and s contains a vertex of P.*

Proof of Fact 1. Assume w.l.o.g. that r contains no vertex of P, and let A_0 and A_1 be the planes tangent to $W(b_0)$ and $W(b_1)$, respectively, that intersect in r. Consider an angle a increasing continuously from b_0 to b_1, and call $A(a)$ the plane that simultaneously changes from A_0 to A_1 and intersects the xy-plane in a line with angle a. At every angle a, $A(a)$ touches P, and, by convexity of P, the set of contact-points is connected. However, since there is no angle of an edge in (b_0, b_1), $A(a)$ cannot touch P in an edge, for $b_0 < a < b_1$. This contradicts either the convexity of P or the assumption that r contains no vertex.

Fact 2. *Let a be an angle in* (b_0, b_1). *Then* $W(a)$ *intersects the xy-plane properly.*

Proof of Fact 2. Let p be any point in Q. By definition of the wedge of a line in the xy-plane, the closure C of the cone $-(C(o)-P(o))+p$ (which has apex p) is contained in the intersection of $W(b_0)$ and $W(b_1)$. Consequently, C contains neither v nor w. For L a line that contains p, the two bounding halfplanes of $W(L)$ are tangent to C. Let $a = a(L)$ be in (b_0, b_1) and let M be a plane orthogonal to L. Since C is contained in the intersection of $W(b_0)$ and $W(b_1)$, the closure of $W(L)^M$ (which equals C^M) is contained in the orthogonal projection of the intersection of $W(b_0)$ and $W(b_1)$ onto M. As a consequence, neither v nor w are contained in $W(L)$, and, moreover, they lie on different sides of $W(L)$. Fact 2 follows, and therewith Lemma 5.4. □

Lemma 5.4 will be used for the design of an efficient algorithm that solves the three-dimensional thickness problem at hand. First, some notation is introduced: Let e be a relevant edge of P and let a be an angle of e. $W(L)$, for L a line with $a = a(L)$, is termed *e-supporting* if there is a translate of $W(L)$ that contains the interior of P and e is contained in a bounding halfplane. A translate W of $W(L)$ is *maximal* if W contains the interior of P and both bounding halfplanes of W support P.

Theorem 5.5. *Let o be a tangible simple object in the xy-plane and let P be a convex polytope with n vertices above the xy-plane. There exists an algorithm that finds in* $O(n \log n)$ *time and* $O(n)$ *space a line L in the xy-plane such that P is in* $W(L)$ (*if it exists*).

Proof. First, an algorithm is outlined that identifies the relevant edges of P and determines their angles. It considers each of the $O(n)$ edges of P in turn and decides their relevance as follows:

Algorithm. Let e be an edge of P with endpoints $P(h_1)$ and $P(h_2)$. Let p denote the intersection of the line through $P(h_1)$ and $P(h_2)$ with the xy-plane. (p is at infinity if h_1 and h_2 have the same size.)

Case 1. p lies inside the closure of h_1 and therefore also inside the closure of h_2. Then e is not relevant.

Case 2. p lies outside the closures of h_1 and h_2. Then compute the lines L_1 and L_2 through p that are tangent to both h_1 and h_2. If $W(L_1)$ is e-supporting, then e is relevant and $a(L_1)$ is an angle of e. Analogously, if $W(L_2)$ is e-supporting, then e is relevant and $a(L_2)$ is an angle of e. If neither $W(L_1)$ nor $W(L_2)$ is e-supporting, then e is not relevant.

Since a plane which contains an edge of P supports P if and only if both faces bounded by the edge lie on the same side of the plane, constant time suffices to check the property of being e-supporting.

Next, for each relevant edge e and angle a of e we determine the maximal translate W of $W(L)$, for $a(L)=a$ and $W(L)$ e-supporting: a determines a plane M perpendicular to L and thus the basic projection of M. Hence, the normals of the bounding halfplanes of W are determined and W is computed by finding the tangent planes of P with these normals. This can be done in $O(\log n)$ time following a method of Dobkin and Kirkpatrick [3].

The final step in the algorithm is to test all maximal translates of wedges. If W is a maximal translate with the closure above the xy-plane, then there is a line L in the xy-plane such that $W(L)$ is a translate of W, and P is contained in $W(L)$. The final step can be accomplished in $O(n)$ time which implies the overall runtime of $O(n \log n)$ as desired. □

With the solution for the special three-dimensional thickness problem we are able to give an algorithm that determines a transversal for a set of homothets (if it exists).

Theorem 5.6. *Let S be a set of n homothets of a tangible simple object in E^2. There exists an algorithm that determines a transversal in* $O(n \log n)$ *time and* $O(n)$ *space (if it exists).*

Proof. In a first step, the convex hull C of $P(S)$ is constructed. Next, the algorithm outlined in the proof of Theorem 5.5 is used to determine a line L in the xy-plane such that $W(L)$ contains C (if L exists). L is a transversal of S by Lemma 5.2. The requirements follow from Proposition 2.2 and Theorem 5.5. □

We close this section with an application of the algorithm outlined in the proof of Theorem 5.5. A trivial modification of it can be used to solve a weighted variant of the statistical problem discussed in Section 4:

Let S be a set of n points in E^2 that are considered to be observations of an affine relationship between x and y. Each point p in S has attached a weight $w(p)$. For a line L we define the *weighted deviation* $\text{wdev}(L) = \max\{w(p)d(p, L) \mid p \text{ in } S \text{ and } d(p, L) \text{ the orthogonal distance of } p \text{ from } L\}$. Find the line that minimizes the weighted deviation.

We leave the proof that $O(n \log n)$ time suffices to solve this problem as an exercise to the interested reader.

6. Discussion

This paper presents a computational study of transversal problems. In particular, sets of axis-parallel rectangles in E^d and sets of translates and homothets of simple objects in E^2 are examined. Also, applications of the methods to problems in computer graphics and in statistics are demonstrated.

We consider the use of geometric transforms that have not been employed for the design of algorithms before as the main contribution of this paper. The reader should consult [1] for an introduction to geometric transforms applied in computational geometry. These transforms lead to a uniform approach to several transversal problems for which not even trivial solutions existed. The efficiency of the obtained algorithms is (to a great deal) due to the efficiency of existing algorithms for convex hull problems and low-dimensional linear programming.

Let us finally mention a few open problems raised by the investigations of this paper. (1) Can the geometric transforms described be exploited to obtain new mathematical insight into transversal problems? (2) Our methods seem to be of value for the computation of a point in the common intersection of a set of geometric objects. This offers alternative and probably more general approaches to common intersection problems (see [1] for other approaches).

Acknowledgment

The author gratefully acknowledges the criticism of an anonymous referee who discovered a serious flaw in an earlier version of this paper.

References

[1] K.O. Brown, Geometric transforms for fast geometric algorithms, Rept. CMU-CS-80-101, Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[2] L. Danzer, B. Gruenbaum and V. Klee, Helly's theorem and its relatives, *Proc. Symp. on Pure Math. VII: Convexity* (1961) pp. 101–180.

[3] D.P. Dobkin and D.G. Kirkpatrick, Fast detection of polyhedral intersections, *Proc. 9th Internat. Coll. on Automata, Languages and Programming*, Lecture Notes in Computer Science **140** (Springer, Berlin, 1982) pp. 154–65; also in: *Theoret. Comput. Sci.* **27** (3) (1983) 241–253.

[4] H. Edelsbrunner, H.A. Maurer, F.P. Preparata, A.L. Rosenberg, E. Welzl and D. Wood, Stabbing line segments, *BIT* **22** (1982) 274–281.

[5] H. Edelsbrunner, M.H. Overmars and D. Wood, Graphics in Flatland: A case study, in: F.P. Preparata, ed., *Advances in Computing Research Vol. 1* (JAI Press, London, 1983) pp. 35–59.

[6] D.E. Field, Algorithms for drawing simple geometric figures on raster devices, Ph.D. Thesis, Dept. Electrical Engineering & Computer Science, Princeton Univ., NJ, 1983.

[7] R.L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inform. Process. Lett.* **1** (1972) 132–133.

[8] R.H. Gueting, Conquering contours—efficient algorithms for computational geometry, Ph.D. Thesis, Dept. of Computer Science, Univ. of Dortmund, 1983.

[9] H. Hadwiger and H. Debrunner, *Kombinatorische Geometrie in der Ebene*, Monogr. de L'Enseignement Mathematique (Geneva, Italy, 1959).

[10] E. Kreyszig, *Statistische Methoden und Ihre Anwendungen* (Vandenhoeck and Ruprecht, Göttingen, 1975).

[11] N. Megiddo, Linear programming in linear time when the dimension is fixed, *J. ACM* **31** (1984) 114–127.

[12] J. O'Rourke, An on-line algorithm for fitting straight lines between data ranges, *Comm. ACM* **24** (1981) 574–578.

[13] F.P. Preparata and S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, *Comm. ACM* **20** (1977) 87–93.

[14] M.I. Shamos, Computational geometry, Ph.D. Thesis, Dept. Computer Science, Yale Univ., New Haven, 1978.