The Visual Computer

Translation separability of sets of polygons

Frank Dehne
and Jörg-Rüdiger Sack *

School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6

We consider the problem of separating a set of polygons by a sequence of translations (one such collision-free translation motion for each polygon). If all translations are performed in a common direction the separability problem so obtained has been referred to as the uni-directional separability problem; for different translation directions, the more general multi-directional separability problem arises. The class of such separability problems has been studied previously and arises e.g. in computer graphics and robotics. Existing solutions to the uni-directional problem typically assume the objects to have a certain predetermined shape (e.g., rectangular or convex objects), or to have a direction of separation already available. Here we show how to compute all directions of uni-directional separability for sets of arbitrary simple polygons.
The problem of determining whether a set of polygons is multi-directionally separable had been posed by G.T.. Toussaint. Here we present an algorithm for solving this problem which, in addition to detecting whether or not the given set is multi-directionally separable, also provides an ordering in which to separate the polygons. In case that the entire set is not multi-directionally separable, the algorithm will find the largest separable subset.

Key words: Computational geometry – Data structures – Motion problems – Separability

* Research supported by NSERC under grant No. A9173 and A0392, respectively

1 Introduction

Motion problems are manifold due to the variety of areas in which they may occur; among these areas we find e.g. robotics, computer graphics, computer vision etc. One class of motion problems recently being investigated is the *separability problem.* To state separability problems formally, we introduce some terminology. For a survey article on separability problems see Toussaint (1985).
Let $\mathbb{P} = \{P_1, \ldots, P_M\}$ be a set of M n-vertex polygons (in the Euclidean plane), with pairwise non-intersecting interiors. A translation of a polygon $P_i \in \mathbb{P}$ is specified by the translation direction and distance. A *separating motion* of P_i is a translation of P_i in some direction by an arbitrarily large distance. P_i is said to *collide* with polygon P_j, $i \neq j$, if, at any distance during the *separating motion*, the interiors of P_i and P_j intersect; otherwise, P_i and P_j are *separable* in the given direction. P_i and P_j *interlock* if there exists no direction in which they are separable. A polygon P_i *is separable from the set*, $\mathbb{P}$, if there exists some direction d such that, in this direction, P_i is separable from each of the remaining polygons P_j, $j \neq i$, $1 \leq j \leq M$. E.g. polygon 1 in Fig. 1 is separable from the polygon set in direction d, as indicated.
A set of polygons $\mathbb{P} = \{P_1, \ldots, P_M\}$ is *sequentially separable* (by a sequence of M translations) if there exists a *multi-directional translation ordering* $(P_{\pi(1)}, P_{\pi(2)}, \ldots, P_{\pi(M)})$ of $\mathbb{P}$ such that for $i = 1, \ldots, M-1$ polygon $P_{\pi(i)}$ is separable from the set of remaining polygons $\mathbb{P}_{\pi(i+1)} = \{P_{\pi(i+1)}, \ldots, P_{\pi(M)}\}$ by a translation in some direction d_i. We refer to this problem as the *Multi-Directional Separability Problem* (MDS-Problem).
In case that all translations are to be performed in a common direction d we obtain the *Uni-Directional Separability Problem* (UDS-Problem) and the respective translation ordering is called a *(uni-directional) translation ordering* for $\mathbb{P}$. A set $\mathbb{P}$ exhib-

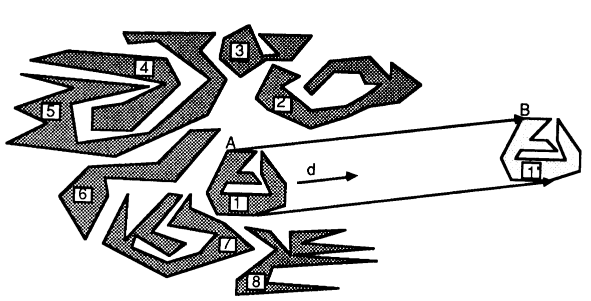

Fig. 1. Polygon 1 can be separated from the object set via a translation in direction d

its the *translation ordering property* if a translation ordering exists in each direction.
In the following section we will review some results from the existing literature some of which we will use in subsequent sections. It turns out that existing solutions to the uni-directional problem assume the objects to have a certain predetermined shape (e.g. rectangular or convex objects), or to have a direction of separability available. In Sects. 3 and 4 we will show how these drawbacks can be overcome. Among other results, we show how to compute all directions of uni-directional separability for sets of arbitrary simple polygons.
In Sect. 5 we will present a solution for the multi-directional separability problem as posed by G.T. Toussaint. In addition to detecting whether or not a given set is multi-directionally separable, the algorithm also provides an ordering among the polygons to perform such a separation. In case the entire set is not multi-directionally separable it finds the largest separable subset.
Summarizing, this paper presents algorithms for the following detection and determination problems (for any set $\mathbb{P}$ of simple polygons in the Euclidean plane with no pairwise intersection occurring among (the polygons):

Detection Problems
- Detect whether a translation ordering for $\mathbb{P}$ exists (in some arbitrary direction).
- Detect whether $\mathbb{P}$ is uni-directionally sequentially separable in a given direction.
- Detect whether $\mathbb{P}$ is multi-directionally sequentially separable.

Determination Problems (UDS-Problem)
- Find a direction in which $\mathbb{P}$ is uni-directionally sequentially separable.
- Determine the set of *all* directions in which $\mathbb{P}$ is uni-directionally sequentially separable.

Determination Problems (MDS-Problem)
- Determine a multi-directional translation ordering for $\mathbb{P}$, including one or all collision-free translation directions for each polygon.
- Determine the maximal subset of $\mathbb{P}$ which is multi-directionally separable.

2 Some results from the literature

Several authors studied the question of determining a translation ordering among a set of objects, in a given direction. The approach taken by Guibas and Yao (1980) was designed to work for rectangles and was adapted to work for convex polygons. For the latter case an alternate solution is due to Mansouri and Toussaint (1985). Touissaint also studied other restrictive classes of objects (e.g., spheres), see Toussaint (1984). Neither of these approaches generalizes to arbitrary simple polygons, since the solutions are typically obtained by exploiting some specific properties of the object classes at hand.
Whereas for convex polygons, spheres etc. a translation ordering will exist for every direction specified (see Corollary 2 below), this is clearly no longer true when dealing with arbitrary simple polygons. Thus a preliminary task is to determine whether or not such an ordering exists. In this section we show how to test a given collection of objects for uni-directional separability.
To determine whether or not a collection of M n-vertex polygons $\mathbb{P}=\{P_1, \ldots, P_M\}$ is uni-directionally separable the following result obtained in Toussaint (1985a) is useful:

Theorem 1. *A set of polygons $\mathbb{P}=\{P_1, \ldots, P_M\}$ admits a translation ordering in direction d if, and only if, every pair of polygons, viewed in isolation, is separable with a single translation in direction d.*

Corollary 2. *A translation ordering will always exist if each pair P_i, P_j of polygons in $\mathbb{P}$ has non-intersecting convex hulls.*

In view of this theorem the study of separability for single pairs of polygons becomes important.
Recall that *two polygons* are *separable* (*by a single translation*) if one of them can be translated an arbitrary distance away from the other without a collision occuring between the objects. Two objects *interlock* if they are not separable. These notions are illustrated in Fig. 2a, b.
The problem of detecting whether two polygons are separable in a *given* direction of translation (described by a vector in the unit circle) can be solved in time, linear in the number of vertices, see (Toussaint and Sack 1983). The solution is based on the observation that two polygons are separable with a single translation in a given direction d if and only if the interior of the visibility hulls for P and Q with respect to the direction orthogonal to d do not intersect. The *visibility hull* of a polygon P in a direction d' is defined as the set obtained by taking the union of P with all line

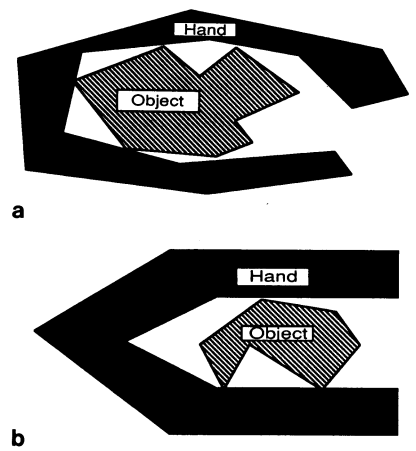

Fig. 2. **a** Hand and object interlock. **b** Hand and object are separable

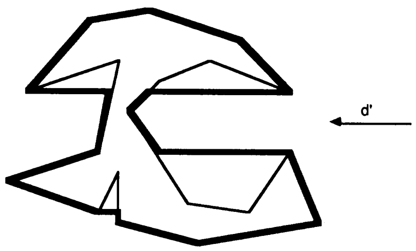

Fig. 3. Visibility hull of a polygon

segments $[a, b]$ parallel to d' with a, b in P; see Fig. 3. Since two visibility hulls can be constructed and subsequently intersected in linear time (they are monotone in the same direction), the following lemma has been obtained (Toussaint and Sack 1983).

Lemma 3. *The problem of detecting whether two polygons can be separated in a* given direction *can be solved in time, linear in the number of vertices of both P and Q.*

In dealing with collections of objects involving more than two simple, pairwise non-intersecting polygons different types of problems arise (for a survey the reader is referred to (Toussaint 1985a)). Toussaint states two algorithms for the problem of detecting whether a collection of M, n-vertex, non-pairwise intersecting polygons is separable in a given direction. His results are based on Lemma 3 together with existing solutions to certain geometrical problems. If each pair of visibility hulls is tested for intersection an $O(M^2 n)$ algorithm is obtained. Alternatively, the M visibility hulls can be considered as a set of $M{*}n$ line-segments and using the method of Shamos and Hoey (1976) a complexity of $O(M n \log M n)$ is derived.

The entire approach using visibility hulls has two *draw-backs*:

(1) The visibility hulls depend on the specified direction and must therefore be *computed for every direction of translation.* In particular, for solving repeatedly separability queries on the existence of a translation ordering, in a given query direction, more efficient algorithms must be designed.

(2) Since there may be an *infinite number of directions of separability*, i.e., directions in which the set is separable, the method cannot be used
(a) to solve the problem of whether the given set is uni-directional separable or not, i.e., it does not solve the uni-directional separability problem
(b) nor to find all directions of separability, if any.

In Nurmi (1984) the same complexity results are obtained by using a plane-sweep technique. However, his approach has the same disadvantages as stated above.

Next it is demonstrated how these drawbacks can be overcome.

3 Some tools for solving separability problems

3.1 Movability wedge for a pair of polygons, movability wheel of a set of polygons

In Sack and Toussaint (1983) the problem of determining *all* directions of separability for two polygons P, Q was addressed. Clearly, if no such direction exists then P and Q interlock. Their approach is based on the observation that if two distinct directions of separability exist then these directions determine an entire wedge of directions of separability. The maximal such wedge, is called the *relative movability wedge* $W_P(Q)$ for P relative to Q. The wedge is maximal in the sense that all directions inside the wedge define directions of separability for P relative to Q and no direction outside the wedge is a direction of separability. The union

of $W_P(Q)$ and $W_Q(P)$ is called the *movability wedge* for P and Q, denoted by $W(P, Q)$.
If we assume that both polygons have the same number of vertices, say n, then the computation of the movability wedge can be performed in $O(n^2)$ time. By combining several tools of computational geometry with a partitioning technique developed for solving this problem, it has been shown in Sack and Toussaint (1985) that this time can be reduced to $O(n \log n)$. Let $C_s(n)$ denote the time to compute the movability wedge for two n-vertex polygons.

Lemma 4. *For two arbitrary n-vertex polygons P and Q. The relative movability wedge $W_P(Q)$ and the movability wedge $W(P, Q)$ can be computed in time $C_s(n) = O(n \log n)$ time.*

Movability wedges have been computed for other more restricted classes of polygons; see Toussaint and Sack (1983), Toussaint and ElGindy (1984), Toussaint (1984, 1985). In Table 1 we summarize all results obtained.
Very recently it has been shown (Sack 1986, unpublished notes; and independently Toussaint 1986, personal communication) that given a triangulation of the vertex set the movability wedge can be computed in linear time.
Movability wedges capture information essential to our solution to separability problems for polygons thereby enabling us to state simple and efficient solutions to uni-directional separability problems.
We introduce the concept of movability wheel for a set $\mathbb{P}$ of polygons. The set of all directions d for which a translation ordering exists is called the *movability wheel* $W(\mathbb{P})$ for $\mathbb{P}$ and for a given direction d we denote by $\mathbb{T}(d)$ the set of all translation orderings of $\mathbb{P}$ with respect to d. We say that d is a *direction of separability* if d is in $W(\mathbb{P})$. We denote by $\mathbb{W}$ the set of all *pairwise movability wedges*, $\{W(P_i, P_j) \mid 1 \leqq i < j \leqq M\}$.

Table 1. Movability wedges for polygon pairs

Movability wedge construction times for pairs of n-vertex polygons

Polygon class	*Complexity*
Constant size	$O(1)$
Convex	$O(\log n)$
Monotone, star-shaped	$O(n)$
Arbitrary simple	$O(n \log n)$

Lemma 5. *For the movability wheel $W(\mathbb{P})$ the following holds:*
(a) *$W(\mathbb{P})$ is the intersection of all pairwise movability wedges $W(P_i, P_j)$ in $\mathbb{W}$.*
(b) *$W(\mathbb{P})$ consists of at most $M(M-1)$ disjoint sectors.*

Proof. (a) follows from Theorem 1.
(b) Let $\mathbb{W}$ denote the set of all pairwise movability wedges. Since $|\mathbb{W}| = M(M-1)/2$ it suffices to show that $W(\mathbb{P})$ consists of at most $2|\mathbb{W}|$ sectors. The result is proved by induction on the cardinality of the set $\mathbb{W}$, called m. For $m=1$ the result follows from the above. Assume that, for $m>1$, the intersection, W', of $m-1$ pairwise movability wedges has at most $2(m-1)$ sectors. Let W_m be the next movability wedge to be intersected with W'. In worst case (with respect to the number of sectors in the intersection of W' with W_m), $2(m-1)-2$ sectors are contained in W_m and the two sectors defined by $[0, 360) - W_m$ are contained in the remaining two sectors. Thus the number of sectors in the intersection of W' and W_m is at most $2(m-1)-2+4=2m$. See Fig. 4 for an illustration.

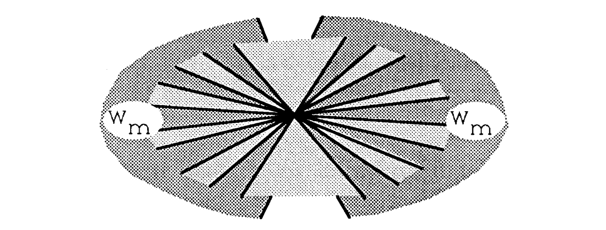

Fig. 4. Worst case intersection of the m^{th} movability wedge W_m

In the following section we introduce a data structure called "complementary segment tree" which allows efficient computation of the intersection of intervals and allows queries of the type "is a given direction contained in the intersection, or not?", as well as "find a direction which is contained in the intersection".

3.2 Manipulating sets of intervals: the complementary segment tree

Let $\mathbb{S} = \{I_1, \ldots, I_k\}$ be a set of k intervals. Such a set of intervals can be stored in a data structure called segment tree, as described e.g., in Mehlhorn (1984). A segment tree is composed of a search part, its internal nodes, and of a data part, its leaves storing the intervals.

Stored with each node v of a segment tree is

(a) an interval $x\,\text{range}(v)$ designed as the union over all intervals stored in the leaves of the subtree rooted at v, and

(b) a list $NL(v) = \{I \in \mathbb{S} \mid x\,\text{range}(v) \text{ is in } I \text{ but } x\,\text{range}(\text{parent}(v)) \text{ is not in } I\}$.

Operations on segments trees are: *Construct*, *Insert* and *Delete*. The *Construct* operations involves setting up the search part, essentially placing a balanced binary search tree on the interval set as well as calculating the $x$range values and the sets NL. The operations, *Delete* and *Insert*, delete or insert an interval, respectively, thus updating the node fields on the path from the interval to the root.

Note that for each I in $\mathbb{S}$, the query "is I in $NL(v)$?" can be answered in $O(1)$ time provided that both $x\,\text{range}(v)$ and $x\,\text{range}(\text{parent}(v))$ are given.

For our application we do not need to explicitly store the lists NL, since storing their sizes, $|NL(v)|$, is sufficient. This reduces the storage from $O(k \log k)$ to $O(k)$ for a segment tree on k intervals. We will, however, need an additional bit, called *mark*(v), stored at each node v. We call a node *marked* if its mark bit is true and *unmarked*, otherwise. The bit is set as follows:

- If v is a leaf then
  mark(v) = true, if $|NL(v)| = 0$
  mark(v) = false, otherwise.
- If v is an internal node then
  mark(v) = true, if $|NL(v)| = 0$ and at least one child v' is marked
  mark(v) = false, otherwise.

We will call such a tree a *modified segment tree* for $\mathbb{S}$. With Mehlhorn (1984) it is easy to prove the following:

Lemma 6. *A modified segment tree for a set $\mathbb{S}$ of k intervals can be constructed in time $O(k \log k)$ time using $O(k)$ space. An interval I in $\mathbb{S}$ can be deleted from the segment tree, i.e., the values $|NL(v)|$ and mark(v) can be updated, in time $O(\log k)$, for all v on the two paths from the root to the end points of the interval.*

Consider now a set $\mathbb{S} = \{w_1, \ldots, w_k\}$ *of k movability wedges and let* $w_i^c := [0, 360) - w_i$. Each w_i^c consists of at most 3 intervals; the set $\mathbb{S}^c$ of all such intervals thus consists of at most $3k$ intervals. The modified segment tree on $\mathbb{S}^c$ is called the *complementary segment tree* for the interval set $\mathbb{S}$. In the case of relative movability wedges, each w_i^c has at most 2 intervals and thus $\mathbb{S}^c$ contains at most $2k$ intervals.

Lemma 7. *Let T be the complementary segment tree for a set $\mathbb{S}$ of movability wedges, and let d be a direction in* $[0, 360)$. *Furthermore let* INT$(\mathbb{S})$ *denote the intersection of all wedges in $\mathbb{S}$.*

(a) d is in INT$(\mathbb{S})$ iff all nodes along the path from the root of T to the leaf containing d are marked.

(b) INT$(\mathbb{S})$ is the union of all intervals stored at leaves v, for which all nodes along the path from v to the root are marked.

(c) INT$(\mathbb{S}) \neq \emptyset$ iff the root of T is marked.

Proof. We will use the fact that $\bigcap_{w_i \in \mathbb{S}} w_i = (\bigcup_{w_i \in \mathbb{S}} w_i^c)^c$. Now let $W_{\mathbb{S}} := \bigcap_{w_i \in \mathbb{S}} w_i$.

(a) $\Rightarrow$ Let $d \in W_{\mathbb{S}}$ and let d be contained in the interval stored at leaf v. Assume that there is an unmarked node on the path from the root of T to v then there is some node, say v', on this path for which $|NL(v')| \neq 0$. Thus there exists some $w_i \in \mathbb{S}$ such that $d \in x\,\text{range}(v') \subseteq w_i^c$ and, thus, d is not in $W_{\mathbb{S}}$ which is a contradiction.

(a) $\Leftarrow$ Assume that d is not in $W_{\mathbb{S}}$ and hence $d \in \bigcup_{w_i \in \mathbb{S}} w_i^c$. W.l.o.g. assume that $d \in w_j^c$. Let v be the leaf of T with $d \in x\,\text{range}(v)$ then by the construction of the segment tree $x\,\text{range}(v)$ is in w_j^c. Hence there exists a node v' on the path from v to the root of T for which $w_j^c \in NL(v')$ and thus mark(v') is false.

(b) follows from (a).

(c) $\Rightarrow$ follows immediately from (a).

(c) $\Leftarrow$ If the root is marked then, using induction, it is easy to show that there is at least one path from the root to a leaf of T for which all its nodes are marked. With this (c) then follows from (a).

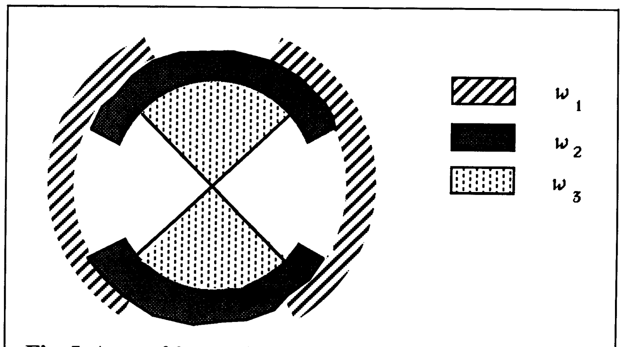

Fig. 5. A set of 3 movability wedges

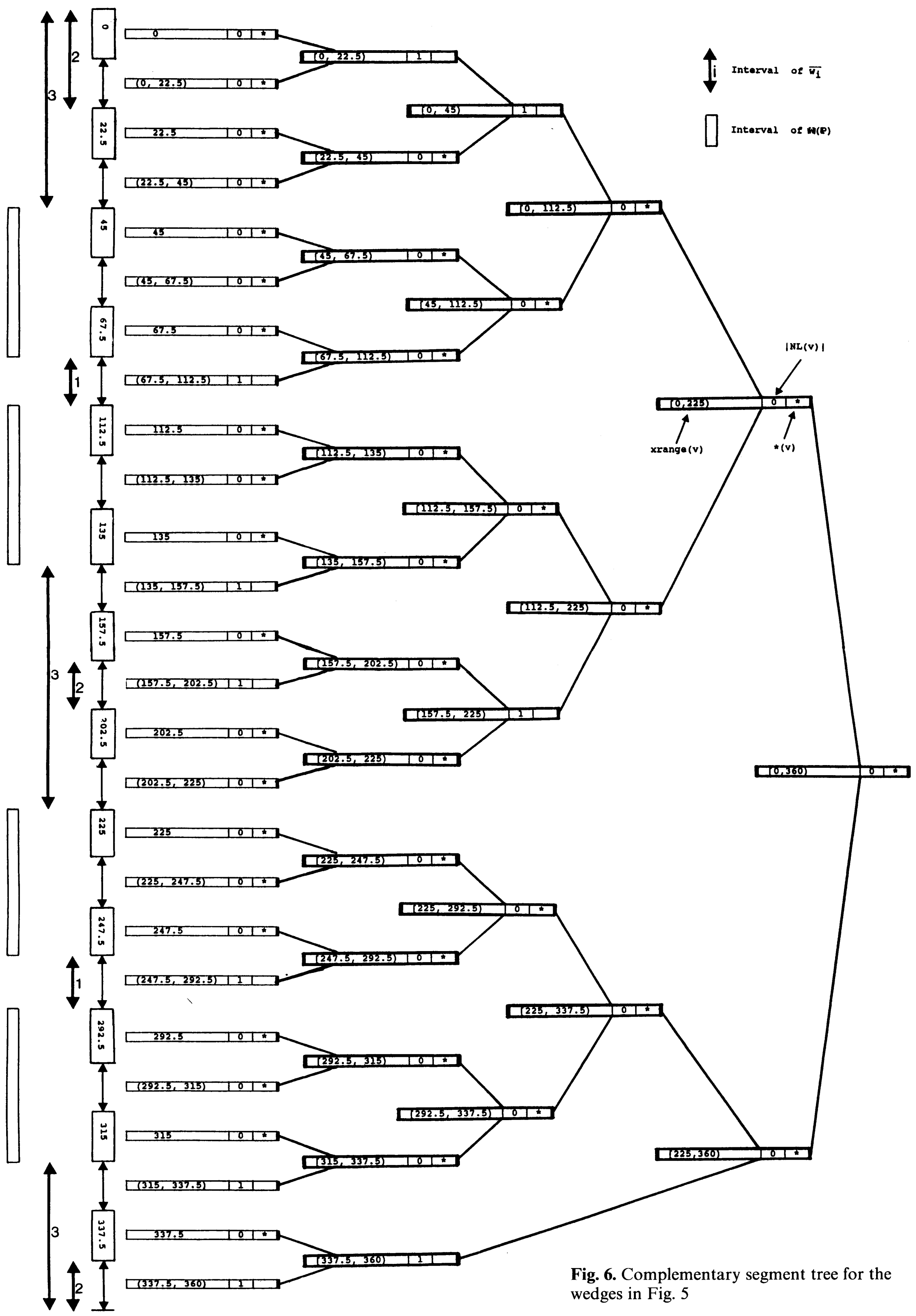

Fig. 6. Complementary segment tree for the wedges in Fig. 5

4 Main results on uni-directional separability problems

4.1 UDS detection

With these results we are now able to solve the UDS detection problems stated above. The movability wheel $W(\mathbb{P})$ contains all directions in which a translation ordering for $\mathbb{P}$ exists. From Lemma 5 we have

$$W(\mathbb{P}) = \bigcap_{W(P_i, P_j) \in \mathbf{W}} W(P_i, P_j).$$

We compute $\mathbf{W}$ in time $O(M^2(C_s(n))$ and the complementary segment tree $T_{\mathbf{W}}$ with respect to $\mathbf{W}$ in time $O(M^2 \log M)$, Lemma 6. Thus, by Lemma 7(c) a translation ordering for $\mathbb{P}$ exists iff the root of $T_{\mathbf{W}}$ is marked and we get

Theorem 8. *The problem of detecting whether any uni-directional translation ordering for $\mathbb{P}$ exists can be solved in $O(M^2(C_s(n)+\log M))$ time.*

Proof. Follows from Lemma 6 and 7(c).

Furthermore, given the complementary segment tree with respect to $\mathbf{W}$ and a direction $d \in [0, 360]$, then we know from Lemma 7(a), that $d \in (W(\mathbb{P}))$ iff all nodes on the path from the root of $T_{\mathbf{W}}$ to the leaf containing d are marked. Since the depth of $T_{\mathbf{W}}$ is $O(\log(M))$, using Lemma 6, we get the following result potentially useful in a dynamically changing environment:

Theorem 9. *Given $O(M^2(C_s(n)+\log M))$ preprocessing, the existence query of a transition ordering with respect to a given direction, for any set of M n-vertex polygons, can be answered in time $O(\log M)$.*

4.2 UDS determination

We will now demonstrate how to compute $W(\mathbb{P})$, once a complementary segment tree $T_{\mathbf{W}}$ has been computed. If the root of $T_{\mathbf{W}}$ is unmarked, then $W(\mathbb{P}) = \emptyset$. Otherwise, assume that $W(\mathbb{P}) \subseteq [0, 360]$ consists of k intervals $I_i, \ldots, I_k$. Since each I_j is contained in all $w_i \in \mathbf{W}$, its interior may not contain the border of any such w_i. Hence, I_j is the union of the x range of at most three leaves of $T_{\mathbf{W}}$. Scanning $3k$ leaves v of $T_{\mathbf{W}}$ to test whether the path from v to the root of $T_{\mathbf{W}}$ is totally marked, takes time $O(k \log M)$. By Lemma 5, k is $O(M^2)$. Finding one leaf v whose $x\,\mathrm{range}(T)$ is in $W(\mathbb{P})$ takes time $O(\log M)$ by taking a totally marked path from the root to some leaf. With this we get:

Theorem 10. (a) *For any set $\mathbb{P}$ of M n-vertex polygons, the set $W(\mathbb{P})$ of all directions d for which a translation ordering of $\mathbb{P}$ exists, can be computed in time $O(M^2(C_s(n)+\log M))$.*
(b) *Given the complementary segment tree $T_{\mathbf{W}}$, then a direction for which $\mathbb{P}$ is uni-directionally sequentially separable can be found in $O(\log M)$ time.*

5 Multi-directional separability

5.1 MDS detection, MDS determination

Recall from the introductory section, that a set of polygons $\mathbb{P} = \{P_1, \ldots, P_M\}$ is *sequentially separable* by a sequence of translations (not necessarily in the same direction), if there exists an ordering O_π, such that for $i = 1, \ldots, M-1$ polygon $P_{\pi(i)}$ can be separated from the remaining polygons $\mathbb{P}_{\pi(i+1)}$, by a translation in some direction d_i. We refer to this problem as the *multi-directional separability* (MDS)-problem, as opposed to the *uni-directional separability problem*, discussed so far, in which all translations are performed in the same direction.
Referring to Fig. 10, while the movability wheel of the polygon set is empty, there exists a sequence of separating motions (in order, polygon 1–6 and finally polygon 7) executed in different directions separating this set. Thus the given polygon set is multi-directionally separable, and it is easily seen that the set is not uni-directionally separable. It should be clear however that uni-directional separability always implies multi-directional separability.
As before, the problem is studied under the two aspects of direction and determination as follows:

- *MDS detection:* Is the given set of polygons multi-directionally separable?
- *MDS determination:* Determine an ordering among the set of polygons.

Our solution to the MDS determination problem will also produce for each polygon to be moved next, the maximal set of directions in which it can be separated.
Again, we use the concept of movability wedges

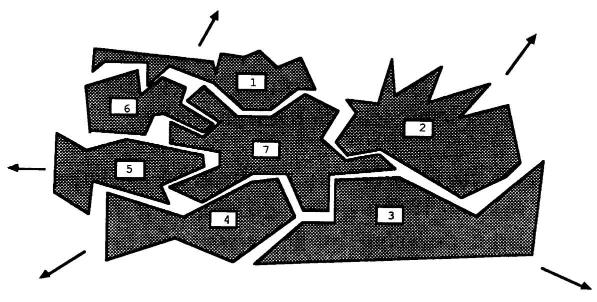

Fig. 7. A multi-directionally separable polygon set which is not uni-directionally separable

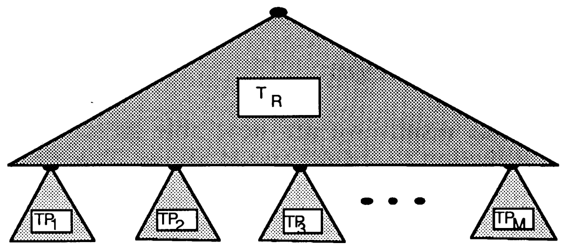

Fig. 8. MDS tree for a polygon set P

in connection with complementary segment trees to efficiently solve these problems.
We denote by $MW(P_i, \mathbb{P}) := \bigcap_{P_j \in \mathbb{P} - \{P_i\}} W_{P_i}(P_j)$ the movability wedge of P_i with respect to $\mathbb{P}$, i.e., the maximum set of directions which admit a translation of P_i without colliding with any $P_j \in \mathbb{P} \setminus \{P_i\}$.
The efficiency of an algorithm to solve MDS problems will depend on how fast (a) it can be determined whether, initially, there exists a polygon P_i which can be separated, i.e., $MW(P_i, \mathbb{P}) \neq \emptyset$ and (b) how fast a next separable polygon can be found after a polygon has been separated from the set. Notice that if a polygon is separable then the movability wedges of the remaining polygons will not decrease as an effect of the separation of the polygon. In particular, this implies that if at any stage of the execution of the algorithm more than one polygon can be separated, the order in which the polygons are separated will have no effect on the decision of whether or not the set is sequentially separable, i.e., on the solution of the MDS detection problem. Our solution will employ the following data structure:
With each polygon $P_i \in \mathbb{P} = \{P_1, \ldots, P_M\}$ we associate a complementary segment tree with respect to the set $\{W_{P_i}(P_j) \mid P_j \in \mathbb{P} \setminus \{P_i\}\}$, called the *wedge-tree* TP_i. In addition to this forest of M wedge trees $TP_1, \ldots, TP_M$ we construct a balanced binary tree, called result tree T_R, whose M leaves are the roots of TP_i. Each internal node v of T_R is marked (i.e., mark(v) is set), if at least one of its sons is marked. Actually, the M wedge trees and the result tree together form a balanced binary tree, which we call the MDS tree of $\mathbb{P}$. See Fig. 8 for an illustration.

With Lemma 6 and Lemma 7 we observe the following:

Property 11

(a) *Polygon P_i is separable from $\mathbb{P}$ if and only if the root of its wedge-tree TP_i is marked.*
(b) *At least one polygon $P_i \in \mathbb{P}$ is separable from $\mathbb{P}$ if and only if the root of T_R is marked.*
(c) *If the root of T_R is marked, then a separable polygon $P_i \in \mathbb{P}$ and its direction of separation can be found in time $O(\log M)$.*
(d) *The MDS-tree of $\mathbb{P}$ can be computed in time $O(M^2(C_s(n) + \log M))$.*

With this, the MDS detection and MDS determination problems can be solved in the following manner:
Initially, the MDS tree of $\mathbb{P}$ is constructed at a cost of $O(M^2(C_s(n) + \log M))$ and, if its root is not marked, then $\mathbb{P}$ is not multi-directionally separable. Otherwise, we find a separable polygon $P_i \in \mathbb{P}$ together with a separating direction $d_i \in [0, 360)$ in time $O(\log M)$. The set of all such directions d_i can be computed in time $O(M \log M)$ in essentially the same way as described in Sect. 5.2.
After P_i has been separated, the MDS tree needs to be updated. This is done by first removing the wedge-tree TP_i and then removing the relative movability wedges $W_{P_j}(P_i)$ from each TP_j, for $j \neq i$. This takes time $O(\log M)$, each, (see Lemma 6) and, thus an accumulated running time of $O(M \log M)$. Finally, T_R is updated in time $O(M)$. This process is iterated at most M times. If $\mathbb{P}$ is multi-directionally separable we obtain a translation ordering for $\mathbb{P}$ together with the set of all translation directions associated with each polygon.

Theorem 12. *Both the MDS detection as well as the MDS determination problem for a set of M n-vertex polygons can be solved in time* $O(M^2(C_s(n) + \log M))$.

Proof. Follows from the above.

5.2 The maximally separable subset problem

In Toussaint (1985) the following problem was posed: If a set of polygons is not sequentially separable, how can a maximally separable subset be determined? The above algorithm solves this problem. This follows since at any time during the execution of the algorithm, all polygons (and only those) whose associated wedge-trees have roots representing non-empty wedges, are separable from the remaining set of polygons. Removing any one of these polygons can never shrink the movability wedges of any other polygon. Thus for the problem of finding the maximally separable subset problem the order in which the polygons are removed is irrelevant. The maximally separable subset is determined when the algorithm encounters a situation in which no more polygons can be removed.

Corollary 13. *The maximally separable subset of a set of M n-vertex polygons can be computed in time* $O(M^2(C_s(n) + \log M))$.

6 Some open problems

The separability problems solved here involve objects in the Euclidean plane. The authors are presently investigating whether an approach similar to the one presented here can be used for solving efficiently separability problems involving objects in 3-spaces.
In this paper separating motions via translations have been studied. Further research is aimed at studying other motions like rotations or screwing motions. Work in this direction can be found in Yap (1983, 1984), Chazelle et al. (1983), Ottmann et al. (1983), Sharir et al. (1986), and Spirakis et al. (1983a, b).

References

Chazelle B, Ottmann T, Soisalon-Soinen E, Wood D (1983) The complexity and decidability of Separation ^{TM}. Tech Rep CS-83-34, Data Structuring Group, University of Waterloo (November 1983)

Guibas, LJ, Yao FF (1980) On translating a set of rectangles. Proc. 12th Ann ACM Symp Theory of Computing, pp 154–160

Mansouri M, Toussaint GT (1985) Translation queries for convex polygons. Proc IASTED Int Symp Robotics and Automation '85, Lugano, Switzerland (June 1985)

Mehlhorn K, (1984) Data structures and algorithms 3: multidimensional searching and computational geometry. Springer, Tokyo Berlin Heidelberg New York, p 212

Nurmi O (1984) On translating a set of objects in 2 and 3 dimensional spaces. Bericht 141, Institut für angewandte Informatik und formale Beschreibungssysteme, Universität Karlsruhe, Federal Republic of Germany

Ottmann T, Widmayer P (1983) On translating a set of line segments. Computer Vision, Graphics and Image Processing 24, pp 382–389

Sack J-R, Toussaint GT (1983) Movability of objects. IEEE Int Symp Inf Theory, St. Jovite, Canada (September 1983)

Sack J-R, Toussaint GT (1985) Translating polygons in the plane. Proc. STACS '85, Saarbrücken, Federal Republic of Germany (January 1985), pp 310–321 (also to appear in Robotica 1987)

Sharir M, Cole R, Kedem K, Leven D, Pollack R, Sifrony S (1986) Geometric applications of Davenport-Schinzel Sequences. Proc 27th Symp Found Comput Sci (Toronto 1986), pp 77–86

Shamos MI, Hoey D (1976) Geometric intersection problems. Proc 7th Ann IEEE Symp Found Comput Sci, pp 208–215

Spirakis P, Yap CK (1983a) Strong NP-hardness of moving many discs. Tech Rep No 92, Courant Institute, NYU (September 1983)

Spirakis P, Yap CK (1983b) On the combinatorial complexity of motion coordination. Tech Rep No 76, Courant Institute, NYU (April 1983)

Toussaint GT (1984) On translating a set of spheres. Tech Rep SOCS-8.4, School of Computer Science, McGill University, Montréal (March 1984)

Toussaint GT (1985) Movable separability of sets. In: Toussaint GT (ed) Computational Geometry. North Holland, pp 335–376

Toussaint GT (1986) Shortest path solves translation separability of polygons. Tech Rep SCS-8.27, School of Computer Science, McGill University, Montréal (October 1985)

Toussaint GT, Sack J-R (1983) Some new results on moving polygons in the plane. Proc Robotic Intelligence and Productivity Conference, Detroit, MI (November 1983), pp 158–163

Toussaint GT, ElGindy H (1984) Separation of two monotone polygons in linear time. Robotica 2:215–220

Whitesides S (1985) Computational geometry and spatial planning. In: Toussaint GT (ed) Computational Geometry. North Holland, pp 377–428

Yap CK (1983) Motion planning for two discs. Tech Rep Courant Institute, NYU

Yap CK (1984) Coordinating the motion of several discs. Tech Rept No 105, Courant Institute, NYU (February 1984)