F.Dehne, J.-R. Sack, "Separeabilty of Sets of Polygons", Proceedings of the 12th International Workshop on Graphtheoretic Concepts in Computer Science (WG'86), June 17-19, 1986, Bernried, W.-Germany, Lecture Notes in Computer Science, Nr. 246, Springer 1986, pp.237-251

SEPARABILITY OF SETS OF POLYGONS

(Preliminary Version)

Frank Dehne and Jörg-Rüdiger Sack*
School of Computer Science
Ottawa, Ont. K1S 5B6
Canada

Abstract

Recently, a growing interest in problems dealing with the movability of objects has been observed. Motion problems are manifold due to the variety of areas in which they may occur; among these areas are e.g. robotics, computer graphics, etc. One motion problem class recently being investigated is the separability problem.

The **separability problem** is as follows: Given a set $\mathbb{P}=\{P_1,...,P_M\}$ of M n-vertex polygons in the Euclidean plane, with pairwise non-intersecting interiors. The polygons are to be separated by an arbitrarily large distance through a sequence of M-1 translations while collisions with the polygons yet to be separated are to be avoided. The **uni-directional separability problem** arises, when all polygons are translated in a common direction; the more general problem of separability through translations in arbitrary directions is referred to as the **multi-directional separability problem.**

Here a simple, novel approach is presented for solving an array of uni-directional and multi-directional separability problems for sets of arbitrary simple polygons. The algorithms presented here provide efficient solutions to these problems and when applied to restricted polygon classes further improvements in the time complexities are achieved.

1. Introduction

To formally state the separability problems discussed in this paper, we introduce some terminology along the lines of a survey article on separability problems [15]. Consider a set $\mathbb{P}=\{P_1,...,P_M\}$ of M n-vertex, simple polygons in the Euclidean plane, with pairwise non-intersecting interiors. A translation of a polygon $P_i \in \mathbb{P}$ is specified by a translation direction and distance. A *separating*

---

* Research supported by NSERC grant No. A0392

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 in $\mathbb{P}$, $i \neq j$, if, at any distance during the *separating motion,* the interiors of P_i and P_j intersect; otherwise, we call P_i and P_j *separable* in the given direction. P_i and P_j are said to *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 P_j, $j \neq i$, $1 \leq j \leq M$. For an illustration see Figure 1 in which polygon 1, at position A, is separated from the polygon set in the indicated direction. Polygons 4 and 5 interlock.

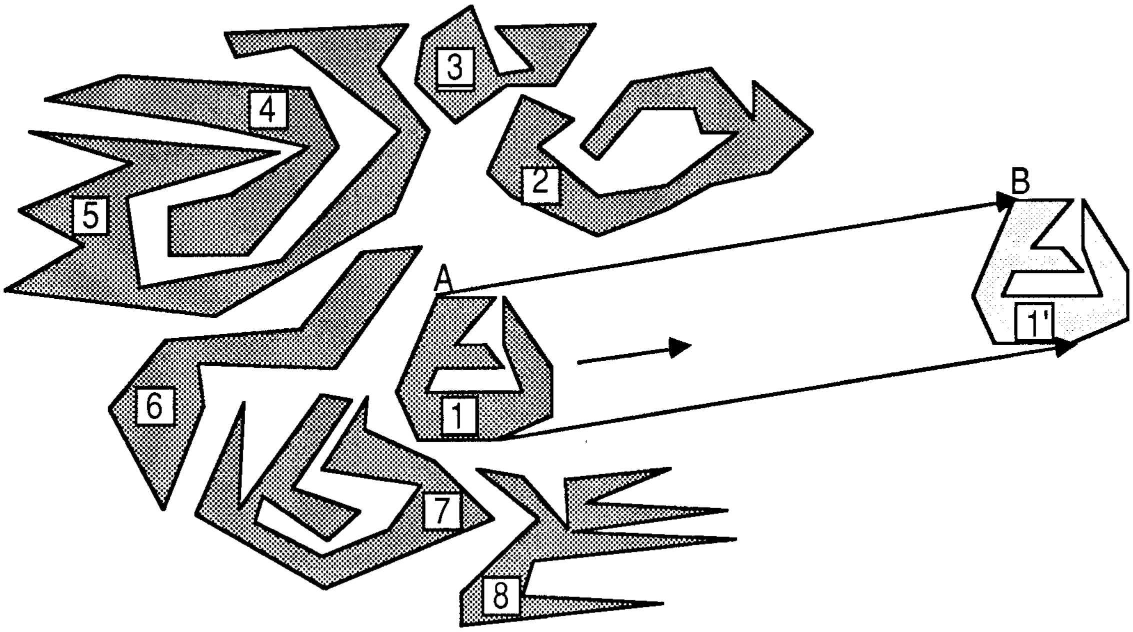

Figure 1: Polygon 1 can be separated from the object set via a translation.

A permutation of the index set {1,...,M} is denoted by π. The ordering among the polygons in $\mathbb{P}$ induced by π is denoted by $\mathbb{O}_\pi$. $\mathbb{P}_{\pi(i)}$ denotes the set of polygons $\{P_{\pi(i)}, P_{\pi(i+1)}, \ldots, P_{\pi(M)}\}$. A set of polygons $\mathbb{P} = \{P_1, \ldots, P_M\}$ is *sequentially separable* (by a sequence of M-1 translations) if there exists an ordering, $\mathbb{O}_\pi$, such that each polygon $P_{\pi(i)}$, i=1,..., M-1, is separable from the set of remaining polygons, $\mathbb{P}_{\pi(i+1)}$ by a translation in some direction d_i. $\mathbb{O}_\pi$ defines an order in which the polygons are separable. Such an ordering is called a *multi-directional translation ordering.*

When studying multi-directional translations different problems arise which we classify as detection and determination problems, and referred to as the <u>*multi-directional separability problems*</u> (MDS-problems).

Detection Problem

- Detect whether $\mathbb{P}$ is multi-directional sequentially separable.

Determination Problem

- Determine a multi-directional translation orderings for $\mathbb{P}$.

For rectangular objects, Guibas and Yao [3] have shown that in some applications the motions of separation are to be performed in a common direction. We refer to resulting problem area for a set of arbitrary simple polygons as the <u>*uni-directional separability problem*</u> *(UDS-problem)* and the ordering, $\mathbb{O}_\pi$, of the polygons in $\mathbb{P}$ is called a (uni-directional) *translation ordering* for $\mathbb{P}$. A set $\mathbb{P}$ exhibits the *translation ordering property* if a translation ordering exists in each direction. We state the following problems:

Detection Problems

- Detect whether a translation ordering for $\mathbb{P}$ exists.
- Detect whether $\mathbb{P}$ is uni-directionally sequentially separable in a given direction.

Determination Problems

- Find a direction in which $\mathbb{P}$ is uni-directionally sequentially separable.
- Determine the set $\mathbb{W}(\mathbb{P})$ of <u>all</u> directions in which $\mathbb{P}$ is uni-directionally sequentially separable.
- For a given direction, determine a uni-directional translation ordering among the objects in $\mathbb{P}$.
- For a given direction d determine the set $\mathbb{T}(d)$ of all orderings of $\mathbb{P}$, so that the objects in $\mathbb{P}$ are uni-directionally sequentially separable in d, when following any of the orderings in $\mathbb{T}(d)$.

Our work on separability problems was originally inspired by the result of Guibas and Yao who studied a uni-directional separability problem for sets of rectangles. They showed that any set of M rectangles possesses the translation ordering property. More importantly, for any given direction the order in which to separate the rectangles can be determined in O(M log M) time. Whereas Guibas and Yao studied primarily sets of rectangles, Chazelle, Ottmann, Soisalon, Wood [1] and Mansouri, Toussaint [4, 12-15] were interested in studying separability problems for sets of less restricted polygon class, such as rectilinear, monotone, convex, or star-shaped polygons. Chazelle et al. showed that certain separability decidability problems can be NP-hard even for rectilinear polygons. Translation problems for line segments were discussed by Ottmann and Widmeyer [7].

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 result is using visibility hulls yielding an O(min(Mn log Mn), M^2n)) time bound. Alternately, using plane sweep techniques, Nurmi [6] obtains an O(Mn log Mn) algorithm for the same problem. His algorithm also generalizes to 3 dimensions. Both approaches [6, 15] have two draw-backs:

(1) They are dependent on the specified translation direction and must therefore be recomputed for every direction of translation. In particular, for solving separability queries on the existence of a translation ordering, in a given query direction, more efficient algorithms can 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.

Here a simple, coherent framework is developed which allows solving an array of separability problems. Unlike previous approaches, the approach presented here is general, in the sense that it provides efficient solutions for sets of arbitrary simple polygons, for restricted polygon classes, as well as for some closed elementary curves like circles, ellipses, etc. The approach is based on the concept of movability wedge, originally introduced in [8]. Movability wedges will be disussed in Section2. In Section 3 we will describe a data structure for solving uni-directional separability problems discussed in Section 4, and for multi-directional separability problems, discussed in Section 5.

2. Movability Wedges

Whereas for convex polygons, spheres etc. a translation ordering will exist for every direction specified (see Corollary 2.2. 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. 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 [15] is useful:

Theorem 2.1 *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 2.2 *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.

2.1 Movability Wedge for Pairs of Polygons

In [9] 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. Notice that the movability wedge of Q with respect to P can be obtained from the movability wedge of P relative to Q by a 180^0 rotation of the wedge. The union of both wedges 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, Sack and Toussaint have shown that this time can be reduced to O(n log n) [9]. Let $C_S(n)$ denote the time to compute the movability wedge for two n-vertex polygons. Very recently, a further improvement has been obtained reducing the complexity $C_S(n)$ to an optimal time of O(n); see [10] for an algorithm based on the above ideas and [16] for an alternate algorithm.

Lemma 2.3 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 linear time.

Movability wedges have also been computed for other more restricted classes of polygons [12-15]; e.g. the movability wedges for a convex polygon can be determined in O(log n) time. Movability wedges capture information essential to our solution to separability problems for polygons thus enabling us to state simple and efficient solutions to uni-directional separability problems.

2.2 Characterization of Movability Wedges

Depending on whether there exists a direction d which is in both relative movability wedges for two polygons P_i and P_j, two different situations arise; these situations are characterized as *Type I and Type II wedges,* respectively; they are defined as:

(I) A movability wedge is of *Type I* if the intersection of both relative movability wedges is empty.

(II) A movability wedge is of *Type II* , otherwise.

In case of a Type II wedge all directions in the intersection of two relative movability wedges for P_i and P_j, allow a separating motion of P_i as well as P_j, irrespective of their order of translation. We therefore call this intersection of relative movability wedges the *irrelevant sectors* of the movability wedge $W(P_i, P_j)$. The *relevant sectors* of the movability wedge are obtained by computing the set difference of the movability wedge minus its irrelevant sectors. Any translation in a direction in the relevant sector requires either P_i or P_j to be moved first. The following properties are easily derived.

Property 2.4 *(a) The movability wedge $W(P_i, P_j)$ is composed of four sectors.*

(b) If a movability wedge $W(P_i, P_j)$ is of Type I then in each direction either no collision-free separation is possible or a unique translation ordering among P_i, P_j is defined.

(c) If the movability wedge $W(P_i, P_j)$ is of Type II then P_i and P_j are separable in any direction of translation; furthermore the relevant sectors define unique symmetrical translation orderings while the two irrelevant sectors define regions of translation for which translation order is irrelevant.

2.3 Common Movability Wedge

We introduce the concept of common movability wedge for a set $\mathbb{P}$ of polygons.The set of all directions d for which a translation ordering exists is called the *common movability wedge* $\mathbb{W}(\mathbb{P})$ 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 $\mathbb{W}(\mathbb{P})$. We denote by $\mathbb{W}$ the set of all *pairwise movability wedges,* $\{W(P_i, P_j) | \ 1 \le i < j \le M\}$.

Lemma 2.5 *For the common movability wedge $\mathbb{W}(\mathbb{P})$ the following holds:*

(a) $\mathbb{W}(\mathbb{P})$ is the intersection of all pairwise movability wedges $W(P_i, P_j)$ in $\mathbb{W}$.

(b) $\mathbb{W}(\mathbb{P})$ consists of at most $M(M-1)$ disjoint sectors.

Proof See the full version of this paper [2].

Thus for solving the UDS-problems we need a structure which allows the intersection of movability wedes to be performed efficiently. The structure, called *inverted segment tree,* will be introduced in the next section; it allows also efficient answers to 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. Manipulating Sets of Intervals: Inverted Segment Tree

Let $\mathbb{S}=\{ I_1, \dots, I_k\}$ be a set of k intervals. The set of intervals can be stored in a data structure called segment tree, as described e.g. in [5]. 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 xrange(v) designed as the union over all intervals stored in the leaves of the subtree rooted at v, and

(b) a list $NL(v) = \{ I \, \varepsilon \mathbb{S} |$ xrange(v) is in I but xrange(parent(v)) is not in $I \}$.

For our application we do not explicitly store the lists NL, but rather store their sizes, $|NL(v)|$. This reduces the storage from $O(k \log k)$ to $O(k)$ for a segment tree on k intervals. We will use 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:

(a) v is a leaf: Then mark(v) is true, if $|NL(v)| = 0$, otherwise marks(v) is false.

(b) v is an internal node: Then mark(v) is true, if $|NL(v)|=0$ and at least one child v' is marked, otherwise mark(v) is false.

We will call such a tree a *modified segment tree* for $\mathbb{S}$.

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 xrange(v) and xrange(parent(v)) are given. With [5, pp. 212] it is easy to prove the following:

Lemma 3.1 *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.*

Consider now a set $W=\{w_1,..,w_k\}$ of k movability wedges linearized on [0, 360) 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 *inverted 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 3.2 *Let T be the inverted 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($\mathbb{S}$) 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 $\bigcap_{w_i \, \varepsilon \, \mathbb{S}} w_i = (\bigcup_{w_i \, \varepsilon \, \mathbb{S}} w_i^c)^c$. Now let $W_{\mathbb{S}} := \bigcap_{w_i \, \varepsilon \, \mathbb{S}} w_i$.

(a) ==> Let d ε $W_{\mathbb{S}}$ 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 $w_i \varepsilon$ $\mathbb{S}$ whose xrange is in w_i^C and thus d is not in $W_{\mathbb{S}}$ which is a contradiction.

<== Assume that d is not in $W_{\mathbb{S}}$ and hence d is in the union over all w_i^C, w_i ε $\mathbb{S}$. W.l.o.g. assume that $d \varepsilon w_j^C$. Let v be the leaf of T with d ε xrange (v) then by the construction of the segment tree xrange(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 ε NL(v') and thus mark(v') is false.

(b) and (c) omitted. q.e.d.

4. Solving UDS Problems

4.1 UDS Detection

With these results we are now able to solve the UDS detection problems stated above. The common movability wedge $\mathbb{W}(\mathbb{P})$ contains all directions in which a translation ordering for $\mathbb{P}$ exists. From Lemma 2.5 we have that $\mathbb{W}(\mathbb{P})$ is the intersection of all $W(P_i,P_j)$ in $\mathbb{W}$ and hence we can compute $\mathbb{W}$ in time $O(M^2(C_S(n))$ and the inverted segment tree T with respect to $\mathbb{W}$ in time $O(M^2 \log M)$, Lemma 3.1. Thus, by Lemma 3.2(c) a translation ordering for $\mathbb{P}$ exists iff the root of $T_{\mathbb{W}}$ is marked and we get

Theorem 4.1 *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 2.5, Lemma 3.1, Lemma 3.2(c). q.e.d.

Furthermore, given the inverted segment tree with respect to $\mathbb{W}$ and a direction $d \varepsilon [0,360]$, then from Lemma 3.2 (a) follows that d ε $\mathbb{W}(\mathbb{P})$ iff all nodes on the path from the root of $T_{\mathbb{W}}$ to the leaf containing d are marked. Since $T_{\mathbb{W}}$ is of depth equal to $O(\log(M))$ and using Lemma 3.1, we get:

Theorem 4.2 Given $O(M^2(C_S(n) + \log M))$ preprocessing, the existence query of a translation 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 Determining directions which admit uni-directional separability

Since, for reasons of efficiency, the inverted segment tree does not store the interval lists NL, explicitly, we must show how to compute $\mathbb{W}(\mathbb{P})$, once an inverted segment tree $T_{\mathbb{W}}$ has been computed. If the root of $T_{\mathbb{W}}$ is unmarked, then $\mathbb{W}(\mathbb{P}) = \emptyset$. Otherwise, assume that $\mathbb{W}(\mathbb{P}) \varepsilon [0,360]$ consists of k intervals $I_1,\dots, I_k$ .Since each I_j is contained in all $w_i \varepsilon \mathbb{W}$, its interior may not contain the border of any such w_i. Hence, I_j is the union of the xrange of at most three leaves of $T_{\mathbb{W}}$. Thus scanning at most 3 k leaves v of $T_{\mathbb{W}}$, and during that process ensuring that the path from v to the root of $T_{\mathbb{W}}$ is totally marked takes time O(klog M). Since k is $O(M^2)$ and finding one leaf v whose xrange(T) is in $\mathbb{W}(\mathbb{P})$ takes time O(log M) we get

Theorem 4.3 *(a) For any set $\mathbb{P}$ of M n-vertex polygons, 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 an inverted segment tree on M^2 movability wedges, a direction for which $\mathbb{P}$ is uni-directionally sequentially separable can be found in O(log M) time.
We have shown how the translation ordering detection problem for any given direction can be solved. Once the existence of such a translation ordering in a given direction d has been established, it remains to be shown how such an ordering can be determined. To accomplish this we first define a graph, MG($\mathbb{P}$,d), called "movability graph" of $\mathbb{P}$ with respect to direction d.

4.3 The Movability Graph

Let d be a direction for which a translation ordering has been established.The *movability graph* MG($\mathbb{P}$,d) of $\mathbb{P}$ with respect to direction d is a directed graph with vertex set P and edge set E defined constructively as follows:

Starting with an empty set of edges we traverse the list of all pairwise movability wedges $W(P_i,P_j)$: For each wedge $W(P_i,P_j)$ we add an edge (P_i,P_j) if d is in a sector where the unique translation ordering is "P_j before P_i" ; an edge (P_j,P_i) if d is in a sector where the unique translation ordering is "P_i before P_j" , respectively. (Such a unique translation ordering occurs if either $W(P_i,P_j)$ is a Type I wedge or i if it lies inside one of the relevant sectors of a Type II wedge.
Note that the graph is not necessarily connected.

Lemma 4.4 *The movability graph MG($\mathbb{P}$,d) can be computed in time $O(M^2 C_s(n))$ and $O(M^2)$ space.*
Proof Lemma 4.4 is an immediate consequence of Lemma 2.3. q.e.d.

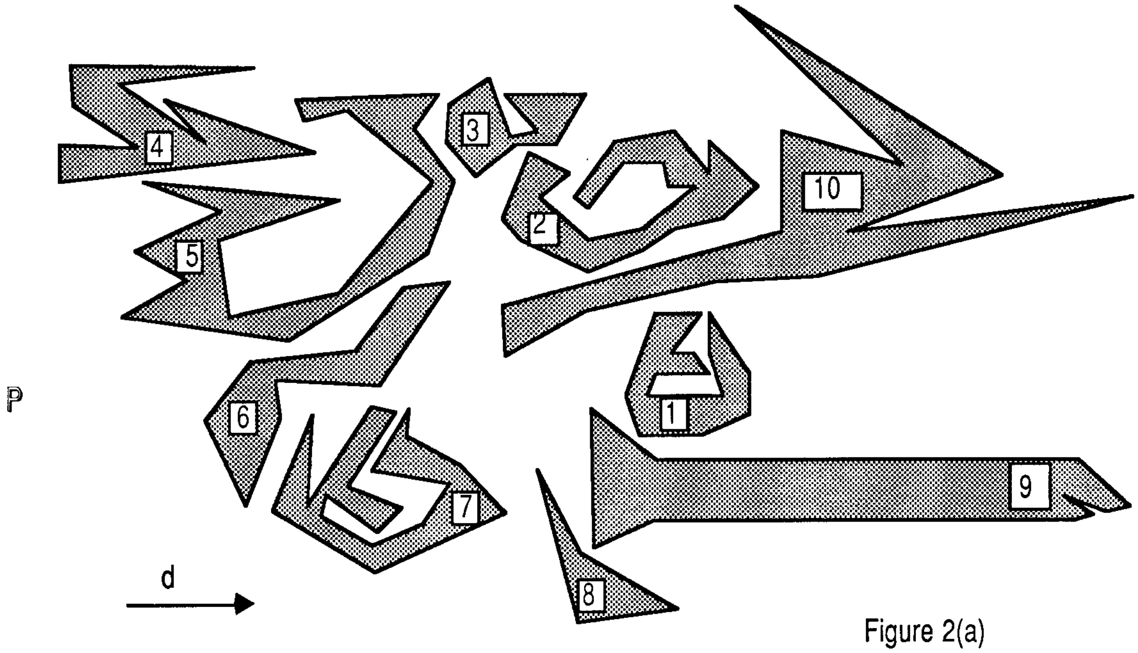

Figure 2(a)

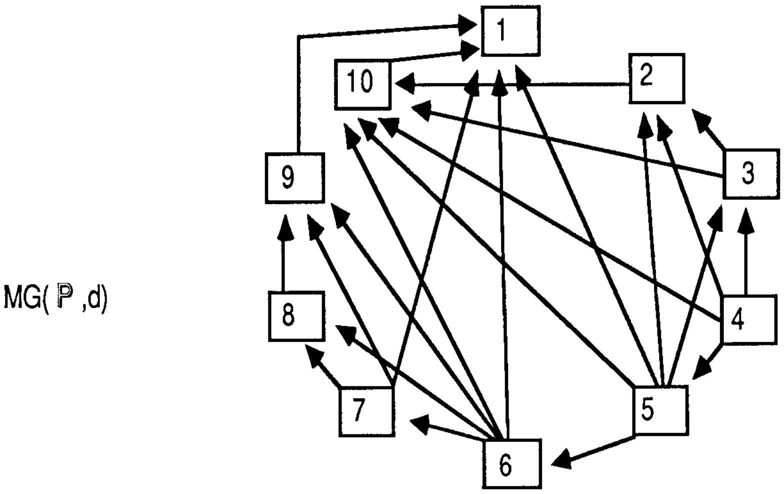

Figure 2(b)

Figure 2: A set $\mathbb{P}$ of polygons, (a) and its movability graph MG($\mathbb{P}$,d) , (b).

4.4 Determining UDS Translation Ordering

We will now show how UDS translation orderings can be determined. Let $P_i =>_d P_j$ denote that there exists an edge (P_i,P_j) in MG($\mathbb{P}$,d) and let $->_d$ be the transitive closure of $=>_d$. If it is clear from the context we will omit the index d (=> instead of $=>_d$). With these definitions we show:

Property 4.5 *If MG($\mathbb{P}$,d) contains no edge (P_i,P_j) then P_i can be translated in direction d without colliding with P_j.*

Proof omitted. q.e.d.

Lemma 4.6 *A permutation π of the index set $\{1,...,M\}$ defines a translation ordering $\mathbb{O}_\pi$ of $\mathbb{P}$ with respect to direction d if and only if there is no pair (i,j), $1 \le i < j \le M$ such that $P_{\pi(i)} \to_d P_{\pi(j)}$.*

Proof Consider any direction d for which there exists at least one translation ordering of $\mathbb{P}$ with respect to d.

"==>" Assume there is a pair (i,j), $1 \le i < j \le M$ such that $P_{\pi(i)} \Rightarrow P_{\pi(j)}$ then we know from the definition of $W(P_{\pi(i)}, P_{\pi(j)})$ that $P_{\pi(i)}$ cannot be moved in direction d before $P_{\pi(j)}$ has been translated . Thus, π does not define a translation ordering.

Assume there is a pair (i,j), $1 \le i < j \le M$ such that $P_{\pi(i)} \to P_{\pi(j)}$ but there is no pair (i',j'), $1 \le i < j \le M$ such that $P_{\pi(i')} \Rightarrow P_{\pi(j')}$, since otherwise the same arguments as above hold. Consider a sequence of polygons $P_{\pi(k1)}, P_{\pi(k2)}, ..., P_{(kt)}$ $(t \ge 1)$ of $\mathbb{P}$ such that $P_{\pi(i)} \Rightarrow P_{\pi(k1)} \Rightarrow P_{\pi(k2)} \Rightarrow ... \Rightarrow P_{(kt)} \Rightarrow P_{\pi(j)}$. Since we assumed that there is no pair (i',j'), $1 \le i < j \le M$ such that $P_{\pi(i')} \Rightarrow P_{\pi(j')}$ we get $j \le k_t \le ... \le k_1 \le i$, a contradiction.

"<==" Assume there is no (i,j), $1 \le i < j \le M$ such that $P_{\pi(i)} \to P_{\pi(j)}$. In order to show that π induces a translation ordering of $\mathbb{P}$ we prove that $P_{\pi(i)}$ can be separated from $\mathbb{P}_{\pi(i+1)}$ in direction d for i=1,..,M. Consider any polygon $P_{\pi(j)}$ in $\mathbb{P}_{\pi(i+1)}$ then j>i. By assumption MG($\mathbb{P}$,d) contains no edge $(P_{\pi(i)}, P_{\pi(j)})$ since otherwise $P_{\pi(i)} \to P_{\pi(j)}$. Thus, by Property 4.5, $P_{\pi(i)}$ can not collide with $P_{\pi(j)}$. q.e.d.

As a consequence of Lemma 4.6 we obtain

Theorem 4.8 *If there exists at least one translation ordering of $\mathbb{P}$ with respect to a given direction d then the set $\mathbb{T}(d)$ of all translation orderings of $\mathbb{P}$ with respect to direction d is exactly the set of all topological sortings of P with respect to MG($\mathbb{P}$,d).*

The maximum length of all directed paths in MG($\mathbb{P}$,d) which start at P in $\mathbb{P}$ will be denoted by D(P,MG($\mathbb{P}$,d)) or D(P) if MG($\mathbb{P}$,d) is clear from the context.The values D(P) for all polygons P can be derived as output of a topological sorting process [see e.g. 11]. Let $\mathbb{P}_0, ..., \mathbb{P}_s$ be the partitioning of $\mathbb{P}$ into disjoint subsets such that $\mathbb{P}_i = \{P \in \mathbb{P} \,/\, D(P, MG(\mathbb{P}, d)) = i\}$ (see Figure 3) and let $\Pi(\mathbb{P}_i)$ denote the set of all permutations of the polygons contained in $\mathbb{P}_i$.

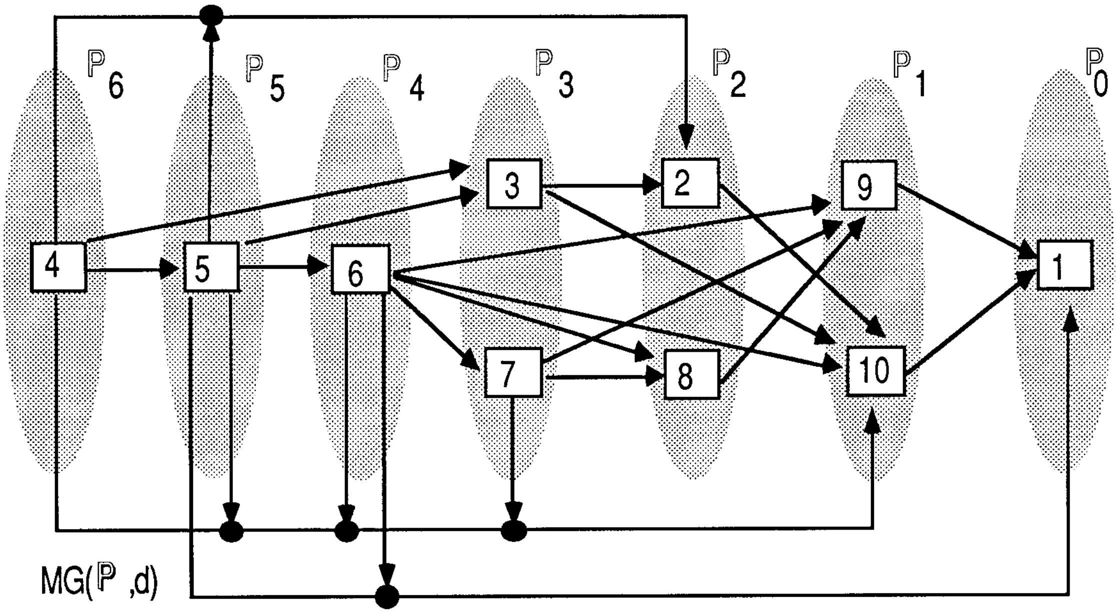

Figure 3: The Movability Graph of Figure 2 in Topological Sorted Order.

By Theorem 4.7 a subset of translation orderings is directly obtained from the graph:

Corollary 4.8 *For a given a direction d of separability* $\Pi(\mathbb{P}_0) \times \Pi(\mathbb{P}_1) \times \ldots \times \Pi(\mathbb{P}_S)$ *is a subset of the set of all possible translation orderingsT(d).*

If all translation orderings are to be computed we associate with each node in the graph a field indicating its outdegree. All polygons (and only those) whose corresponding nodes have zero outdegree are separable in direction d. After separation of such a polygon its corresponding node is deleted and outdegree fields are updated accordingly. Based on this a procedure for computing all possible translation ordering is easily written.

The graph also reflects which polygons and in which order a given non-separable polygon would encounter on a translation movement. These polygon will have to be moved aside to clear the path for the given polygon.

5. The Multi-Directional Separability Problem

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 $\mathbb{O}_\pi$, such that for i=1,...,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.

It should be clear that uni-directional separability implies multi-directional separability. However, the converse may not necessarily be true. The

interested reader is invited to construct an example.
Again we use the concept of movability wedges inconnection with the inverted segment trees to efficiently solve multi-directional separability problems.

We denote by $MW(P_i,\mathbb{P}) := \bigcap_{P_j \varepsilon \mathbb{P} - \{P_i\}} W_{P_i}(P_j)$ 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 $P_j \varepsilon \mathbb{P} \backslash \{P_i\}$.

The efficiency of an algorithm to solve the MDS-problems will depend on how fast (a) it can be determined whether, initially, there exists a polygon P_i which can be separated $MW(P_i,\mathbb{P}) \neq \emptyset$ and (b) how fast a new separable polygon can be found once a polygon has been separated from the set. Notice that if a polygon is separable then the movability wedges for all remaining polygons can only increase after the separation has been performed. 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 $P_i \varepsilon \mathbb{P}=\{P_1,...,P_M\}$ we associate an inverted segment tree for the set $\{W_{P_i}(P_j) \mid P_j \varepsilon \mathbb{P} \backslash \{P_i\}\}$, called the*wedge-tree* TP_i. In addition to this forest of M wedge trees $TP_1,...,TP_M$ we construct a balanced binary tree, called result tree T_R , whose M leaves are the roots of TP_i. Each interval 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 $\mathbb{P}$. With Lemma 3.1 and Lemma 3.2 we observe the following:

Property 5.1 *(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 $P_i \varepsilon \mathbb{P}$ 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 $P_i \varepsilon \mathbb{P}$ 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 $O(M^2 (C_S(n) + \log M))$ if its root is not marked, then $\mathbb{P}$ is not multi-directionally separable. Otherwise, we find a separating polygon $P_i \varepsilon \mathbb{P}$ together with a separating direction $d_i \varepsilon [0, 360)$ in time $O(\log M)$. (The set of all such directions d_i could be computed in time $O(M \log M)$ as described in Section 4.1.) After P_i has been separated the MDS-tree has to be updated.This is done by first removing the wedge-treeTP_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 3.1) and, thus we get an accummulated running time of $O(M \log M)$. Finally, T_R is updated in time $O(M)$.The entire 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 translation directions associated with each polygon.

Theorem 5.2 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. q.e.d.

Maximally Separable Subset Problem

In [15] 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 also 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 all wedge-trees have unmarked roots thus no more polygons can be removed.

Some Open Problems

The separability problems solved here involve objets in the Euclidean plane. In remains open whether an approach similar to the one presented here can be used for solving efficiently separability problems involving objects in 3-space.

In this paper we have studied separating motions via translations. E.g. for automatic generation of exploded pictorials of part assemblies other separating motions, like rotations, or screwing motions might be considered.

The movability graph, sorted in topological order, can be stored in $O(M^2)$ space and can be generated in $O(M^2 C_S(n))$ time. Since there may be an exponential number of translation orderings just listing these requires at least the same time. However, it is, to the best of the authors' knowledge, an open problem whether the number of such orderings can be generated in a more efficient manner, i.e. in polynomial time. The problem is equivalent to .determining the number of linear extensions of a poset.

References

[1] Chazelle, B., T. Ottmann, E. Soisalon-Soinen, and D. Wood, "The complexity and decidability of Separation^{TM} ", Tech. Rept. CS-83-34, Data Structuring Group, University of Waterloo, November 1983.
[2] Dehne, F. and J.-R. Sack, "Separability of Sets of Polygons", Tech. Rept. SCS-82, School of Computer Science, Carleton University, Ottawa, Oct. 1985.
[3] Guibas, L.J. and F.F. Yao, "On translating a set of rectangles", *Proceedings 12th Annual ACM Symposium on Theory of Computing* , 1980, pp. 154-160.
[4] Mansouri, M. and G.T. Toussaint, "Translation queries for convex polygons", *Proceedings IASTED International Symposium on Robotics and Automation'85,* Lugano, Switzerland, June 1985.
[5] Mehlhorn, K., *Data Structures and Algorithms 3: Multidimensional Searching and Computational Geometry,* Springer Verlag, Heidelberg,1984.
[6] Nurmi, Otto, "On translating a set of objects in 2 and 3 dimensional spaces", Bericht 141, Institut für angewandte Informatik und formale Beschreibungsverfahren, Universität Karlsruhe, Karlsruhe, Federal Republic of Germany.
[7] Ottmann, T. and P. Widmayer, "On translating a set of line segments", *Computer Vision, Graphics and Image Processing 24,* 1983, pp. 382-389.
[8] Sack, J.-R. and G.T. Toussaint, "Movability of objects", *IEEE International Symposium on Information Theory,* St. Jovite, Canada, September 1983.
[9] Sack, J.-R. and G.T. Toussaint, "Translating polygons in the plane", *Proceedings STACS'85,* Saarbrücken, Federal Republic of Germany, January 1985, pp. 310-321.
[10] Sack, J.-R., "A linear-time algorithm for computing separability of pairs of polygons", unpublished notes, Carleton University, Ottawa, 1986.
[11] Sedgewick, R., *Algorithms* , Addison-Wesley, Reading MA, 1983.
[12] Toussaint, G.T. and J.-R. Sack, "Some new results on moving polygons in the plane", *Proceedings Robotic Intelligence and Productivity Conference,* Detroit, MI., November 1983, pp. 158-163.
[13] Toussaint, G.T. and H. ElGindy, "Separation of two monotone polygons in linear time", *Robotica,* Vol. 2, 1984, pp. 215-220.
[14] Toussaint, G.T., "On translating a set of spheres", Tech. Rept. SOCS-8.4, School of Computer Science, McGill University, Montréal, March 1984.
[15] Toussaint, G.T. "Movable separability of sets", in *Computational Geometry* , Ed. G.T. Toussaint, North Holland, 1985, pp.335-376.
[16] Toussaint, G.T., Privat communication, 1986.