On the Qualitative Structure of a Mechanical Assembly

Randall H. Wilson*
rwilson@cs.stanford.edu

Jean-Claude Latombe
latombe@cs.stanford.edu

Robotics Laboratory
Department of Computer Science, Stanford University
Stanford, CA 94305, USA

Abstract

A mechanical assembly is usually described by the geometry of its parts and the spatial relations defining their positions. This model does not directly provide the information needed to reason about assembly and disassembly motions. We propose another representation, the non-directional blocking graph, which describes the qualitative internal structure of the assembly. This representation makes explicit how the parts prevent each other from being moved in every possible direction of motion. It derives from the observation that the infinite set of motion directions can be partitioned into a finite arrangement of subsets such that over each subset the interferences among the parts remain qualitatively the same. We describe how this structure can be efficiently computed from the geometric model of the assembly. The (dis)assembly motions considered include infinitesimal and extended translations in two and three dimensions, and infinitesimal rigid motions.

Introduction

Many application tasks, such as design, process planning, and robot programming, require reasoning about mechanical assemblies. Typical questions that arise in such tasks are: In which order can a physical device A be assembled or disassembled? How many hands are required? What are all the parts or subassemblies that can be directly removed from A? Is it possible to extract a given subassembly without previously removing some other part? What is the minimal number of parts that should be removed prior to the extraction of a specified subassembly? These are, for example, the kind of questions an autonomous maintenance robot in a space station would have to routinely answer in order to plan (dis)assembly operations for diagnosing failures and repairing hardware equipment on board the station. In a different domain, a smart CAD system would help design products that are easier to manufacture and maintain by answering such questions. For instance, while a new product is being designed, it could verify that this product (at its current stage of definition) can be assembled with simple motions and that all critical parts are easily removable for inspection and/or repair.

An assembly A is usually represented as a geometric model, which describes the individual parts composing A and the spatial relations among them. However, this model does not directly provide the information that would allow us to easily answer the above questions. We propose another representation of A, in the form of a qualitative structure that explicitly describes how the parts prevent each other from being removed from A along every possible direction of motion. This representation, the *non-directional blocking graph*, or NDBG, makes it possible to efficiently answer these questions. We show how it can be computed from the original geometric model of the assembly.

The notion of an NDBG derives from the remark that the infinite set of motion directions can be partitioned into an arrangement of subsets, called *regular regions*, such that over each region the interferences among the parts remain qualitatively the same. The boundary between two regular regions consists of *critical* directions where one or more interferences change suddenly; for example, a part P_1 may block the motion of a part P_2 along any direction in one region, but along none in the other region. In each regular region, a directed graph, called the *directional blocking graph*, or DBG, represents the interference relations among the parts of the assembly for any direction of motion in the region. The set of all regular regions, their adjacency relation, and the corresponding DBG's together form the NDBG of the assembly.

The NDBG is a good example of a qualitative representation of a physical device obtained by discretizing a continuous set (the set of motion directions for an object) into a finite number of classes (the regular regions) based on the identification of physical criticalities (the boundaries of the regular regions), such that

*This research was funded by DARPA contract N00014-88-K-0620 (Office of Naval Research), the Stanford Integrated Manufacturing Association, and Digital Equipment Corporation.

one can associate a constant symbolic structure (the DBG) to every class. We will show that the size of the NDBG is polynomial in the number of parts in the assembly and the complexity of these parts, and that it can be computed in polynomial time from the original geometric model of the assembly. Our algorithm computes the DBG for one regular region, and then uses this DBG to derive the DBGs in the adjacent regions by applying a simple *crossing rule*. The transitive closure of this crossing rule yields the full NDBG.

In the next section we review previous related work. In succeeding sections we define the notions of DBG and NDBG for the simplest case of infinitesimal translations, and describe the computation of the NDBG. Finally, we extend the notion of an NDBG (and its computation) to infinitesimal motions in translation and rotation, and to extended translations. This paper combines and extends previous results presented in [Wilson, 1991; Wilson and Matsui, 1992; Wilson and Schweikard, 1992].

Related Work

Reasoning about mechanical assemblies has attracted the interest of AI researchers for a long time. E.g., see BUILD [Fahlman, 1974], NOAH [Sacerdoti, 1977], and RAPT [Popplestone *et al.*, 1980].

It plays a key role in assembly sequence planning, a topic which has recently been under active study. See [Homem de Mello and Lee, 1991] for a collection of papers on this topic. The concept of local freedom of translation used in the definition of a DBG was previously introduced in this context [Homem de Mello, 1989]. The constraints on the feasible infinitesimal motions in translation and rotation of a part, imposed by a contact with another part, are analyzed in [Hirukawa *et al.*, 1991; Ohwovoriole, 1980] yielding the extension of the NDBG to infinitesimal generalized motions.

Related work in Computational Geometry includes methods for separating sets [Toussaint, 1985]. Given an assembly A in the plane, the problem of finding a subassembly $S \subset A$ removable from the rest of A by a single translation is addressed in [Arkin *et al.*, 1989]. An algorithm to construct a sequence of translations separating two polygons is given in [Pollack *et al.*, 1988]. The minimum number of hands needed to (dis)assemble a given device is investigated in [Natarajan, 1988]. See also the work in motion planning [Latombe, 1991].

The notion of an "aspect graph" used in Computer Vision has the same qualitative flavor as the NDBG. The aspect graph of an object describes the appearance of the object from all possible viewpoints. Aspect graphs were first computed by discretizing the set of orientations. Recent algorithms take advantage of the fact that, except at critical orientations, the occluding contours of an object remain qualitatively (i.e. topologically) the same for small changes in the viewpoint (e.g., [Kriegman and Ponce, 1990]).

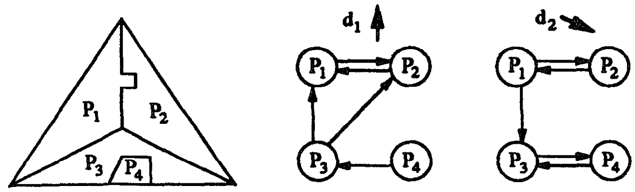

Figure 1: A simple assembly and two DBGs

Qualitative reasoning in continuous domains is a major topic in AI [de Kleer and Williams, 1991]. It requires discretizing a space into regions that are then treated as single entities. Often this discretization is not based on any sort of criticality (discontinuity, singularity, or event) between regions. When criticalities are identified, as in this paper, they yield a more meaningful discretization. A subdomain, qualitative kinematics [Joskowicz and Sacks, 1991], studies the internal motions of parts in an operational device.

Directional Blocking Graph

We consider an assembly A made of n parts $P_1, \ldots, P_n$. We assume that each part is described as a bounded connected regular[1] subset of $\mathbf{R}^2$ (2D case) or $\mathbf{R}^3$ (3D case). The interiors of any two parts in A are disjoint. If the boundaries of two parts intersect, the two parts are said to be in contact. The assembly may or may not be connected.

Suppose that we wish to remove one part, say P_i, by translating it along a direction defined by the unit vector d. A part P_j of A ($j \neq i$) *blocks* the translation of P_i along d iff an arbitrarily small translation of P_i along d leads the interiors of P_i and P_j to intersect. Hence, if P_j blocks P_i, the two parts are necessarily in contact. A subassembly S of A, i.e. a subset of its parts, is *locally free in direction d* iff no part in $A \backslash S$ blocks the translation of any part of S along d.

The *directional blocking graph* of A for an infinitesimal translation along d, denoted by $G(d, A)$, is defined as follows. The nodes of $G(d, A)$ represent the parts P_1 through P_n. An arc of $G(d, A)$ connects P_i to P_j iff P_j blocks the translation of P_i along d. Fig. 1 shows a simple assembly of 4 polygonal parts and its DBGs for two directions d_1 and d_2.

A subassembly S of A is locally free in direction d iff no arcs in $G(d, A)$ connect parts in S to parts in $A \backslash S$. If $G(d, A)$ is strongly connected, no such subassembly exists. Otherwise, at least one strong component of $G(d, A)$ must have no outgoing arcs. For example, in Fig. 1 the subassemblies $\{P_1, P_2\}$ and $\{P_1, P_2, P_3\}$ are locally free in direction d_1; only the subassembly $\{P_3, P_4\}$ is locally free in direction d_2.

For a subassembly S to be removable by a translation along d without previously removing other parts of

[1] A subset of a topological space is *regular* iff it is equal to the closure of its interior.

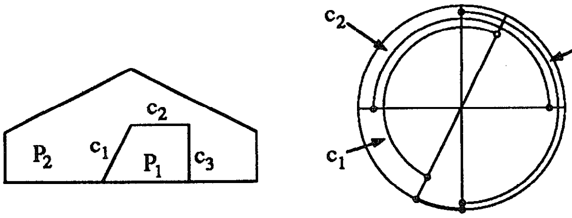

Figure 2: Blocking directions for two parts in contact

A, it must be locally free in direction d. This condition is necessary but not sufficient. Also, the removal of S may require rotation. For those reasons, we will extend the notion of an (N)DBG to infinitesimal motions with rotation and non-infinitesimal motions below.

Non-Directional Blocking Graph

From now on, we will assume that all the parts in the assembly A are polygonal (2D case) or polyhedral (3D case). We will also assume that the intersection of any two parts of A either is empty or consists of straight contact segments of non-zero lengths (2D case) or pieces of planar contact surfaces of non-zero areas (3D case). Both assumptions could be retracted, but at the cost of lengthening our presentation without bringing much additional insight. For the first assumption, we could accept parts bounded by algebraic curves (or surfaces) of any degree; this would seriously complicate the computation of the NDBGs. For the second, we could allow two parts to touch, say, at an isolated point; this would require us to explicitly consider additional, mostly straightforward, particular cases. These cases are treated in [Wilson, 1992].

2D case We represent the set of all translation directions by the unit circle S^1. This circle is the locus of the extremity of the vector d when its origin is attached at the center of S^1.

Let P_i and P_j be two parts in contact, whose intersection consists of the contact segments $E_1, \ldots, E_q$ (Fig. 2). For each segment E_k, we draw the diameter of S^1 parallel to E_k. The two endpoints of this diameter partition S^1 into an open half-circle circle I_k and a closed half-circle $S^1 \backslash I_k$, with the open half-circle I_k containing the direction pointed by the outer normal to P_i in E_k. Hence, for all directions in I_k, P_j blocks P_i. The union $J = \bigcup_{k=1}^{q} I_k$ forms one open connected circular arc, or two non-intersecting open half-circles, or the full circle. The complement of J in S^1 is a closed circular arc, or a single point, or two antipodal points, or the empty set. It contains all the directions d along which P_i is locally free to translate relative to P_j.

For every contact segment in the intersection of two parts of A, we draw the diameter parallel to that segment. The endpoints of all the diameters and the open circular arcs between them form a partition of S^1. We regard each element of this partition as a subset (possibly, a singleton) of directions. The directional blocking graph $G(d, A)$ remains constant when d varies over any such subset. Indeed, by construction of the partition, if a part P_j blocks another part P_i for any direction in the subset, then P_j also blocks P_i for any other direction in the same subset. Any subset R in the partition of S^1 is called a *regular region* and we denote the DBG of A for any direction in R by $G(R, A)$.

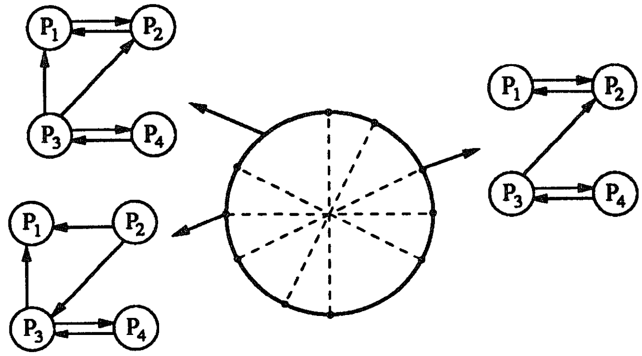

Figure 3: Part of the NDBG of the assembly of Fig. 1

Let $(R_1, \ldots, R_p)$ be the list of regular regions on S^1, with R_k and $R_{k+1 \,(\mathrm{mod}\, p)}$ $(k \in [1, p])$ adjacent. The *non-directional blocking graph* of A is $\Gamma_t(A) \equiv ((R_1, G(R_1, A)), \ldots, (R_p, G(R_p, A)))$. It represents the blocking structure of A for infinitesimal translations.

Part of the NDBG of the assembly of Fig. 1 is shown in Fig. 3. The partition of S^1 in this example consists of 20 regular regions.

3D case We represent the set of translation directions by the unit sphere S^2. With each contact surface between two parts P_i and P_j we associate a plane parallel to this surface and passing through the center of S^2. This plane cuts S^2 along a great circle that partitions the sphere into an open half-sphere and a closed one; the open half-sphere contains the direction of the outer normal to P_i in the contact surface. For every contact planar surface in the intersection of two parts of A, we draw the corresponding great circle on S^2. The set of obtained circles determines an arrangement of regions of three types: *vertices* lie at the intersection of two or more great circles; *edges* are maximal open connected arcs in great circles not including vertices; and *faces* are maximal open connected pieces of spherical surface not intersecting edges and vertices. By construction, each region of the arrangement is *regular* in the sense that the directional blocking graph $G(d, A)$ remains constant when d varies over the region. Let $G(R, A)$ denote the DBG of A for any direction in R.

Let $\mathcal{R} = \{R_1, \ldots, R_p\}$ be the set of regular regions partitioning S^2. Let $\Delta = (\mathcal{R}, \mathcal{L})$ be the non-directed graph representing the adjacency of these regions. A link of $\mathcal{L}$ connects any two regions R_i and R_j such that the boundary of one contains the other.

The *non-directional blocking graph* of A is $\Gamma_t(A) \equiv (\{R_1, G(R_1, A)), \ldots, (R_p, G(R_p, A)\}, \mathcal{L})$.

Computation of Blocking Graph

We assume that the input consists of the geometric models of the parts in the assembly and the specification of the contact segments (or surfaces) between parts. In many situations the latter information is not given explicitly and must be extracted from input spatial relations among the parts, as in [Wilson, 1992]. Let n be the number of parts in A and c the total number of contact segments (or surfaces). We represent each DBG as an $n \times n$ adjacency matrix.

2D case The partition of S^1 contains $O(c)$ regular regions and is easily computed in $O(c \log c)$ time. For each regular region R, we select a direction d in R and compute the DBG $G(d, A)$. After clearing the adjacency matrix for $G(d, A)$, each contact is considered separately. For a contact between parts P_i and P_j, the inner product of d with the outer normal to P_i in the contact segment with P_j is computed. If the inner product is strictly positive, an arc from P_i to P_j is added to $G(d, A)$ (if it does not already exist). The computation of $G(d, A)$ takes $O(n^2 + c)$ time. By repeating this computation for all regular regions, the NDBG is constructed in $O(cn^2 + c^2)$ time.

This computation can be reduced by noticing that there is little or no change between the DBGs of two adjacent regions. This leads to computing the DBG for one region, call it R_1, and then incrementally modifying this graph to get the DBG for the next region in the NDBG list, and so on, until all the regions have been considered. To that purpose, we slightly modify the DBG of a region by attaching a *weight* to each arc of the graph. In $G(R_1, A)$, this weight is the number of inner products that were strictly positive in the above computation. The absence of an arc from P_i to P_j is treated as an arc of weight 0, and conversely. Let R_1 be a circular arc. The next region R_2 in the (circular) NDBG list is necessarily a singleton. Let D be the diameter of S^1 that ends at R_2, and $\{E_1, \ldots, E_s\}$, $s \geq 1$, be the set of all contact segments in A parallel to D. $G(R_2, A)$ can be derived from $G(R_1, A)$ by applying the following *crossing rule*: "Initialize G to $G(R_1, A)$. For every contact segment E_k, let P_i and P_j be the two parts sharing this segment. If the inner product of any direction in R_1 and the outgoing normal to P_i in E_k is strictly positive, then retract 1 from the weight of the arc connecting P_i to P_j in G."

The graph G obtained at the end of the loop is $G(R_2, A)$. (Again, every arc weighted by 0 is interpreted as no arc.) If R_1 is a singleton and R_2 a circular arc, the crossing rule is: "Initialize G to $G(R_1, A)$. For every contact segment E_k, let P_i and P_j be the two parts sharing this segment. If the inner product of any direction in R_2 and the outgoing normal to P_i in E_k is strictly positive, then add 1 to the weight of the arc connecting P_i to P_j in G."

The cost of computing the NDBG is the sum of: $O(c \log c)$, the cost of partitioning S^1; $O(n^2 + c)$, the cost of computing the first DBG; $O(cn^2)$, the cost of copying the remaining $O(c)$ DBGs; $O(c)$, the cost of computing the remaining DBGs. [Note: The cost of computing a DBG by applying the crossing rule is proportional to the number s of contact edges involved in the computation. This number is in $O(c)$, but throughout the computation of the entire NDBG, each edge is considered only twice. Hence, the time complexity of the computation of *all* the remaining DBGs is only $O(c)$.] Therefore, the total cost of computing $\Gamma_t(A)$ is $O(c \log c + cn^2)$. The size of $\Gamma_t(A)$ is $O(cn^2)$.

3D case The main difference with the 2D case is in the computation of the arrangement of great circles on the sphere. Each great circle is derived from a contact surface between two parts, so there are c great circles. We project them into a plane tangent to the sphere and not parallel to any of the great circles, using the central projection from the center of the sphere. We obtain an arrangement of c lines in the plane that intersect at $O(c^2)$ points, producing $O(c^2)$ regions. These regions and their adjacency relations can be computed in optimal $O(c^2)$ time using a topological sweep [Chazelle *et al.*, 1985; Edelsbrunner, 1987] and in $O(c^2 \log c)$ time using a simpler line sweep [Preparata and Shamos, 1985]. The arrangement on the sphere is a direct by-product of the computed arrangement in the plane and has the same size. The rest of the computation is similar to the 2D case. The DBG is computed in an arbitrarily selected region of the arrangement. The DBG in an adjacent region is computed using a straightforward adaptation of the crossing rule stated in the 2D case. $\Gamma_t(A)$ is computed in $O(c^2n^2)$ time. Note that the size of $\Gamma_t(A)$ is $O(c^2n^2)$, and thus this algorithm is optimal in the 3D case.

Incremental modification Assume that we modify A by the addition (or deletion) of δ_n new parts generating (resp., suppressing) δ_c contact edges. If $\delta_n \ll n$ and $\delta_c \ll c$, then rather than computing the NDBG of the new assembly from scratch, we can compute it by modifying $\Gamma_t(A)$ at a relatively low cost. For lack of space we will not describe this computation here.

Property The crossing rule between two regions entails the following property of $\Gamma_t(A)$: "Let R_1 and R_2 be two regular regions such that R_1 is in the boundary of R_2. If there exists an arc from P_i to P_j in $G(R_1, A)$, this arc also exists in $G(R_2, A)$. In particular, if $G(R_1, A)$ is strongly connected, so is $G(R_2, A)$." This is useful to more efficiently exploit the NDBG.

Implementation Methods similar to the above (specifically, those described in [Wilson, 1991; Wilson, 1992]) were implemented in LISP on a DECstation

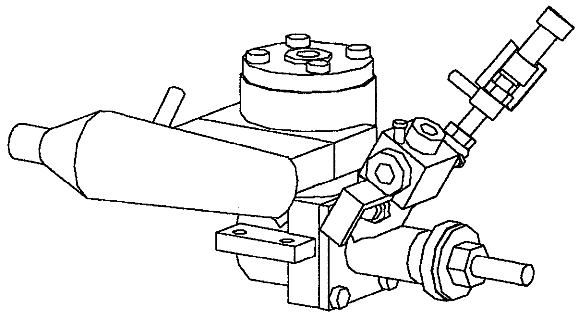

Figure 4: An industrial example

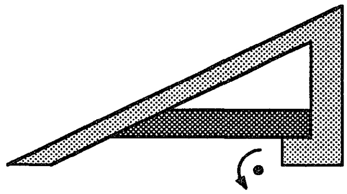

Figure 5: The need for rotation

5000 using floating-point arithmetic. The software has been used to plan assembly sequences for several 2D and 3D products, including a 22-part electric bell and the 42-part engine shown in Fig. 4.

Extension and Variant

Infinitesimal rigid motions One important extension to the NDBG allows motions in rotation. Fig. 5 shows a simple case where one part blocks another for any infinitesimal translation, while a rotation is feasible. Let us consider the 3D case only (the 2D case is just simpler). The direction of an infinitesimal rigid motion is described by a unit vector in a 6D space. Hence, we represent the set of all possible directions of motion by the unit sphere S^5 in $\mathbf{R}^6$. The definition of an NDBG for infinitesimal translations extends to infinitesimal generalized motions in the straightforward way. We denote this new NDBG by $\Gamma_g(A)$.

Consider parts P_i and P_j sharing a piece of planar surface. For each vertex v of the convex hull of the contacting area, the set of all directions of motion that make P_i slide in contact with P_j at v is the intersection of S^5 with a 5-dimensional hyperplane passing through the center of S^5 [Wilson and Matsui, 1992]. This hyperplane partitions S^5 into two half-spheres. The set of hyperplanes determined by the contact surfaces among the parts of A determines an arrangement of regular regions of dimensions 0 (vertex), $1, \ldots, 5$ on S^5. The DBG is constant over each regular region. Since a vertex in the arrangement arises at the intersection of five hyperplanes, the arrangement contains $O(c^5)$ regions, where c is the number of vertices bounding the convex hulls of contacts. It is constructed in $O(c^5)$ time by a multi-dimensional topological sweep [Edelsbrunner, 1987]. Constructing a DBG in any region is done in time $O(cn^2)$. A crossing rule similar to the pure translational case can be established, yielding an $O(c^5n^2)$ time algorithm to build the NDBG for a 3D assembly. Again, since the NDBG is of size $O(c^5n^2)$, this method is optimal.

A necessary condition for a subassembly S to be directly removable from A is that there exists a DBG G in $\Gamma_g(A)$ such that no arcs in G connect parts in S to parts in $A\backslash S$. However, this condition is not sufficient.

Extended translations A useful variant of the NDBG is to consider non-infinitesimal motions. The variant is relatively simple if we restrict motions to *extended translations.* An extended translation is an infinite translation along a single direction $d \in S^1$ (2D case) or S^2 (3D case). Then, given any two parts P_i and P_j in A (these two parts need not be in contact), we say that P_j blocks the translation of P_i along d iff the area that P_i sweeps out when it translates along d from its initial position in A to infinity intersects P_j. The notions of a DBG and an NDBG for extended translations follow from this new blocking relation. We denote the new NDBG by $\Gamma_e(A)$.

Again, consider the 3D case. Let P_i and P_j be any two parts in A. The set $\mathcal{B}$ of directions d along which P_j blocks P_i is identical to the set of directions along which the "grown" object $P_j \ominus P_i = \{a_j - b_i \mid a_j \in P_j, b_i \in P_i\}$ (i.e., the Minkowski difference of the two sets of points P_j and P_i at their position in A) blocks the translation of the origin O [Lozano-Pérez, 1983]. It can be shown that if P_i and P_j are polyhedra, then $P_j \ominus P_i$ is also a polyhedron [Latombe, 1991; Lozano-Pérez, 1983]. Hence, the set $\mathcal{B}$ is the intersection of S^2 and the polygonal cone of all rays erected from O and intersecting $P_j \ominus P_i$. This intersection is a region of S^2 bounded by segments of great circles. The great circles supporting these segments form an arrangement of regions (vertices, edges, and faces). Each region is regular in the sense that the DBG for extended translations remains constant when d varies in it. The arrangement and the associated DBGs form the NDBG of the assembly for extended translations. See [Wilson and Schweikard, 1992] for an algorithm to compute the NDBG for extended translations.

A sufficient condition for a subassembly S to be directly removable from A is that there exists a DBG G in $\Gamma_e(A)$ such that no arcs in G connect parts in S to parts in $A\backslash S$. This condition is not necessary.

The property stated in the previous section for $\Gamma_t(A)$ also holds for $\Gamma_g(A)$ and $\Gamma_e(A)$.

Conclusion

The classical geometric model of a physical device contains enough information to answer questions about the (dis)assembly of the device. However, this information is not explicitly given. We have developed another representation, the non-directional blocking graph, which explicitly describes the internal qualitative blocking structure of the device. We have defined this representation for three types of motions: infinitesimal translations ($\Gamma_t(A)$), infinitesimal generalized motions ($\Gamma_g(A)$), and extended translations ($\Gamma_e(A)$). We have shown how it can be derived from the original geometric model of the assembly.

$\Gamma_t(A)$, $\Gamma_g(A)$, and $\Gamma_e(A)$ can be used to efficiently answer various questions about the (dis)assembly of A. For instance, $\Gamma_e(A)$ can be used to determine whether A can be assembled (or disassembled) with extended translations only. This information can help design for manufacturability, since the cost of manufacturing a product depends critically on the complexity of the motions required by its assembly. If extended translations are not sufficient, $\Gamma_t(A)$ and $\Gamma_g(A)$ can then be used to check whether the device *may* be (dis)assembled with more complex motions. These two NDBGs only provide necessary conditions. If the product satisfies them, a more sophisticated path planner such as the one described in [Barraquand and Latombe, 1991] is needed to give a definitive answer. Such a planner is costly to run. The NDBGs, acting as low-cost filters, considerably reduce the number of times it need be called. In fact, in our experiments on industrial assemblies, NDBGs have supplied the vast majority of part motion calculations required for assembly planning.

The notion of an NDBG as presented in this paper still has a number of shortcomings. For instance, it treats parts as if they were free to fly with arbitrary precision. Grasping, fixturing, tolerancing, and uncertainty are important problems not addressed in our NDBGs. They also assume that every (dis)assembly operation involves the motion of a single subassembly relative to the rest of the assembly. This corresponds to having only two hands (the assembly table, if any, being one). Although a "well-designed" product usually satisfies this constraint, the (dis)assembly of an arbitrary device with n parts may require up to n hands (i.e., it may require every part to move relative to every other part simultaneously) [Natarajan, 1988]. Despite these limitations, we think that NDBGs are clean, useful structures on top of which one may build more complicated ones.

References

Arkin, E. M.; Connelly, R.; and Mitchell, J. S. B. 1989. On monotone paths among obstacles, with applications to planning assemblies. In *Proc. of the ACM Symp. on Computational Geometry.* 334–343.

Barraquand, J. and Latombe, J-C 1991. Robot motion planning: A distributed representation approach. *Int. Journal of Robotics Research* 10(6):628–649.

Chazelle, B.; Guibas, L. J.; and Lee, D. T. 1985. The power of geometric duality. *BIT* 25:76–90.

de Kleer, J. and Williams, B. C. (guest editors) 1991. Special volume, Qualitative Reasoning about Physical Systems II. *Artificial Intelligence* 51(1-3).

Edelsbrunner, H. 1987. *Algorithms in Combinatorial Geometry.* Springer, Heidelberg.

Fahlman, S. E. 1974. A planning system for robot construction tasks. *Artificial Intelligence* 5(1):1–49.

Hirukawa, H.; Matsui, T.; and Takase, K. 1991. A general algorithm for derivation and analysis of constraint for motion of polyhedra in contact. In *Proc. of the Int. Workshop on Intelligent Robots and Systems.* 38–43.

Homem de Mello, L. S. and Lee, S., editors 1991. *Computer-Aided Mechanical Assembly Planning.* Kluwer Academic Publishers, Boston.

Homem de Mello, L. S. 1989. *Task Sequence Planning for Robotic Assembly.* Ph.D. Dissertation, Carnegie Mellon University.

Joskowicz, L. and Sacks, E. 1991. Computational kinematics. *Artificial Intelligence* 51(1–3):381–416.

Kriegman, D. J. and Ponce, J. 1990. Computing exact aspect graphs of curved objects: Solids of revolution. *Int. Journal of Computer Vision* 5(2):119–135.

Latombe, J-C 1991. *Robot Motion Planning.* Kluwer Academic Publishers, Boston.

Lozano-Pérez, T. 1983. Spatial planning: A configuration space approach. *IEEE Transactions on Computers* C-32(2):108–120.

Natarajan, B. K. 1988. On planning assemblies. In *Proc. of the ACM Symp. on Computational Geometry.* 299–308.

Ohwovoriole, M. S. 1980. *An Extension of Screw Theory and its Application to the Automation of Industrial Assemblies.* Ph.D. Dissertation, Stanford University.

Pollack, R.; Sharir, M.; and Sifrony, S. 1988. Separating two simple polygons by a sequence of translations. *Discrete and Computational Geometry* 3:123–136.

Popplestone, R. J.; Ambler, A. P.; and Bellos, I. M. 1980. An interpreter for a language for describing assemblies. *Artificial Intelligence* 14:79–107.

Preparata, F. P. and Shamos, M. I. 1985. *Computational Geometry: An Introduction.* Springer-Verlag.

Sacerdoti, E. 1977. *A Structure for Plans and Behavior.* American Elsevier.

Toussaint, G. T. 1985. Movable separability of sets. In Toussaint, G. T., editor 1985, *Computational Geometry.* Elsevier, North Holland.

Wilson, R. H. and Matsui, T. 1992. Partitioning an assembly for infinitesimal motions in translation and rotation. In *Proc. of the Int. Conf. on Intelligent Robots and Systems.* To appear.

Wilson, R. H. and Schweikard, A. 1992. Assembling polyhedra with single translations. In *Proc. of the IEEE Int. Conf. on Robotics and Automation.* To appear.

Wilson, R. H. 1991. Efficiently partitioning an assembly. In Homem de Mello, Luiz Scaramelli and Lee, Sukhan, editors 1991, *Computer-Aided Mechanical Assembly Planning.* Kluwer Academic Publishers, Boston. 243–262.

Wilson, R. H. 1992. *On Geometric Assembly Planning.* Ph.D. Dissertation, Stanford University.