A Divide-and-Merge Approach to Automatic Generation of Contact States and Planning of Contact Motion*

Jing Xiao and Xuerong Ji
Computer Science Department
University of North Carolina - Charlotte
Charlotte, NC 28223, USA
xiao@uncc.edu and xji@uncc.edu

Abstract

Planning contact motion is important for many robotic tasks but difficult in general due to high variability and geometrical complexity of contact states. It is desirable to decompose the problem into simpler subproblems. A promising decomposition treats the problem as consisting of (1) automatic generation of a discrete contact state graph, and (2) planning contact transitions between neighboring contact states and contact motions within the same contact state. This paper addresses a divide-and-merge approach on solving the general problem by such a decomposition. It discusses issues related to solving the two subproblems and provides examples of automatically generated contact state graphs between two contacting 3-D polyhedra by the approach, which extend the results for 2-D polygons reported in [11].

1 Introduction

From a classic motion planning point of view, planning contact motions means planning motions on the surface of configuration space obstacles (C-obstacles) [20]. The key is to know the C-obstacles. However, most of the work in the literature on computing C-obstacles from given physical objects/robots are limited to 3-D C-obstacles (i.e., C-obstacles of 2-D objects) [2, 3, 22, 23], and only a few approximate C-obstacles of 3-D objects [8, 15]. Computing 6-D C-obstacles exactly for general 3-D polyhedra remains a formidable open problem. However, contact motions, unlike collision-free motions, require *exactness.* Hence, a recent trend is to explore contact motion planning without explicitly computing C-obstacles [9].

Moreover, information of *high-level, discrete* contact states is often most crucial for devicing compliant control strategies to implement contact motions [4, 24], where each high-level state captures the topological and physical contact characteristics often *common to* two or more contact configurations. For instance, the contact state of "a coffee mug sitting on a table" means the bottom surface of the mug contacting the top surface of the table, which is shared by infinitely many mug configurations relative to the table. Indeed, many fine motion or assembly planning systems require the knowledge of high-level, discrete contact states [1, 5, 6, 7, 16, 21, 22, 25, 26]. However, such high-level information is difficult to extract from low-level contact configurations, i.e., C-obstacle surfaces, only.

Out of the necessity to simplify the problem and the need to know high-level, discrete contact states, we decompose the general contact motion planning problem into two subproblems:

Subproblem 1: automatic generation of a discrete contact state graph[1], and

Subproblem 2: planning contact motions between two known adjacent contact states.

Subproblem 2 can be further decomposed to (a) planning an instantaneous contact transition (between the two adjacent contact states) and (b) planning contact motions within the *same* known contact state, which is a *lower-dimensional* and of *smaller-scope* motion planning problem.

We use a divide-and-merge approach to attack the above subproblems, which is characterized by directly exploiting both topological and geometrical knowledge of contacts in the *physical space* of objects. In Section 2, we review the notion of *contact formation* (CF)

*This research is supported by the National Science Foundation under grant IIS-9700412.

[1]With such a graph, one can further plan a sequence of contact state transitions by graph search.

in terms of *principal contacts* (PC) [27] to characterize topological contacts and define contact states as CF-connected regions of contact configurations. We also examine the neighboring relations among contact states and characterize the contact state space as a contact state graph. In Section 3, we describe our divide-and-merge approach towards solving both subproblems listed above and discuss related issues. We present some implementation results for 3-D polyhedra in Section 4 and conclude the paper in Section 5.

2 Contact State Graph

In this section, we first review concepts used to characterize topological contacts and describe their geometrical interpretations. Then, we discuss connectivity of topological contacts and define contact states and a contact state graph.

2.1 Principal Contacts and Contact Formations

Denoting the *boundary elements* of a face as the edges and vertices bounding it, and the boundary elements of an edge as the vertices bounding the edge, A *principal contact* (PC) is defined as the contact between a pair of surface elements (i.e., faces, edges, or vertices) which are not the boundary elements of other contacting surface elements (if there is more than one pair in contact) [27]. There are ten types of PCs between arbitrary polyhedra, as shown in Figure 1. Each non-degenerate PC is associated with a *contact plane*, defined by a contacting face or the two contacting edges in an e-e-cross PC. Each degenerate PC is characterized as between two convex edge or vertex elements and without being associated with a contact plane[2].

With the notion of PC, an arbitrary contact between arbitrary polyhedra is described as the set of PCs formed, called a *contact formation* (CF).

The *geometrical representation of a PC* (GeoPC) denotes the set of contact configurations which satisfy the contact condition described by the PC's topological definition. The *geometrical representation of a CF* (GeoCF) denotes the set of contact configurations which satisfy every contact condition represented by every PC in the CF. Obviously, GeoCF is the intersection of the GeoPC's of the PCs defining the CF.

[2]Note that a contact between a convex edge/vertex and a concave edge/vertex is regarded not as a single PC but as consisting of a couple non-degenerate PCs.

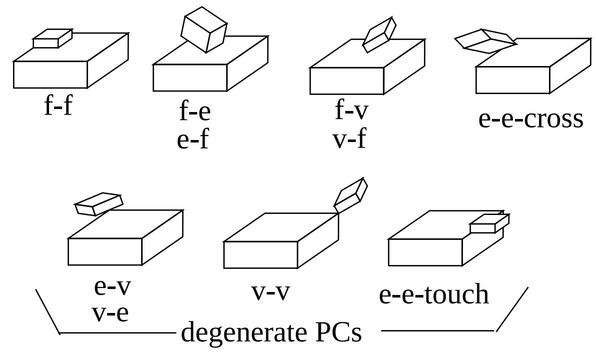


Figure 1: Principal Contacts (PCs)

2.2 Connectivity of CFs and Contact States

Within a contact formation CF, the GeoCF of CF may constitute either a single connected region or multiple connected regions of configurations. In the latter case, from a configuration in one *CF-connected* region to that of another, there is no *CF-compliant* motion, i.e., a motion constrained by the same CF. On the other hand, within a CF-connected region, there exists a CF-compliant motion from one configuration to any other configuration. Thus, we can define a *contact state* between two polyhedra as a single CF-connected region, represented by the CF and a representative configuration in the region, denoted as a pair $<CF, C>$.

With the above definition, the distinction between two contact states with the same CF lies in the two different contact configurations representing two different CF-connected regions.

Now we consider connectivity between contact states of different CFs, $<CF_i, C_i>$ and $<CF_j, C_j>$. If, from C_i to C_j, there exists a path of only configurations in $<CF_i, C_i>$ succeeded by configurations in $<CF_j, C_j>$, then $<CF_i, C_i>$ and $<CF_j, C_j>$ are *generally-defined neighboring contact states*, and CF_i and CF_j are called *generally-defined neighboring contact formations*.

As CFs characterize discrete contact states *topologically*, it can be shown that the above configuration-based definition of neighboring CFs can be mapped to a topological one in terms of how PCs and CFs are related [11]. PC_j a *less-constrained neighbor* (LCN) PC of PC_i, if and only if PC_j is obtained by changing one contacting element of PC_i to one of its boundary elements. Conversely, PC_i is the *more-constrained neighbor* (MCN) of PC_j. For example, in Fig. 1, the e^A-f^B PC is the LCN of the f^A-f^B PC (assuming the objects are A and B) if e^A is a boundary edge of f^A.

Subsequently, CF_j is an LCN CF of CF_i, if and only if (1) CF_j has no more PCs than CF_i, and (2) for every PC in CF_j, it either belongs to CF_i or is an LCN PC of a unique PC in CF_i, and no two PCs in CF_j are LCN PCs of the same PC in CF_i. Conversely, CF_i is the MCN CF of CF_j.

A contact state $<CF_j, C_j>$ is a LCN (or MCN) of another contact state $<CF_i, C_i>$, if and only if these two contact states are generally-defined neighboring contact states and CF_j is a LCN (or MCN) CF of CF_i.

The contact state space (of the contacting objects) can be defined as a contact state graph, where each node denotes a contact state $<CF, C>$, and each arc connects two neighboring contact states.

3 Divide-and-Merge Approach

As introduced in Section 1, we decompose the contact motion planning problem into **Subproblem 1** and **Subproblem 2**. The two subproblems are quite related and can be tackled by a unified divide-and-merge approach.

3.1 Divide and merge contact state graphs

We divide the contact state graph into special subgraphs, called *Goal-Contact Relaxation* (GCR) graphs [30], and further transform the above **Subproblem 1** into *automatically generating the GCR graphs and merging the GCR graphs.* A GCR graph is defined with respect to a "goal" contact state **g** and include all the LCN contact states of **g**, their subsequent LCN contact states, and so on; thus, it can be generated or grown from **g** (the seed node) through repeatedly "relaxing" contact constraints to reach LCNs and add them to the GCR graph until there is no further LCN[3]. Such a seed **g** is a locally most-constrained contact state and often indicates a goal or an intermediate goal state of an assembly.

As shown in [11], which addresses the generation of GCR graphs for contacting polygons, a GCR graph is much easier to generate automatically than an arbitrary contact state graph because it is easier to determine an LCN contact state from a given contact state $\mathbf{s} = \mathbf{CF_i}, \mathbf{C_i}$. This is also true for contacting 3-D polyhedra. First, a possible LCN CF of CF_i can be

[3]This means that a GCR graph is naturally bounded, where the sink or leaf nodes are the least constrained contact states

hypothesized (*unlike* MCN CF) from CF_i based on the topological definition of a LCN CF. Then, the problem becomes one of checking if there exists a feasible contact motion from **s** to the hypothesized LCN which is compliant to no other CF than the two involved. If so, a valid LCN contact state is found; otherwise, hypothesize another and check again. This process essentially heads towards solving **Subproblem 2** stated above.

A large contact state graph can be created by merging GCR graphs. The entire contact state graph is obtained if all GCR graphs are merged. There are two ways to combine multiple GCR graphs. One is to do it sequentially by growing a new GCR graph to meet an existing graph (which could consist of one or more GCR graphs) where there are shared states. Another is to generate each GCR graph independently and then merge the GCR graphs in a separate phase. The key issue in automatic merge is to determine if two nodes from the two graphs respectively represent the same contact state and should be merged into one node in the combined graph. Clearly, only the nodes sharing the same CF can possibly represent the same contact state. Therefore, the problem becomes: given two contact states $<CF_i, C_1>$ and $<CF_i, C_2>$, where $C_1 \neq C_2$, whether there exists a CF_i-compliant motion connecting C_1 and C_2. This, again, is part of what **Subproblem 2** addresses.

3.2 Divide and merge contact motions

A contact motion between only two neighboring contact states can be further divided into two types of motions:

- CF-compliant *reconfiguration* motion, which is finite and changes contact configurations within the same CF, and
- infinitesimal *transition* motion which changes a configuration in one CF to one in the neighboring CF.

Thus, the previously stated **Subproblem 2** can be reduced, again, into *planning CF-compliant reconfiguration under a known CF* and *planning infinitesimal transition* between two neighboring CFs.

Depending on the type and geometry of the CF, objects constrained by a CF can either be fixed or have at most 5 degrees of freedom (DOF) for CF-compliant motion. Usually the more PCs a CF has, the fewer DOF has the CF-compliant motion[4]. This compensating nature (i.e, higher DOF with simpler CF, and

[4]Except for cases with *redundant* PCs, such as those with parallel contact planes.

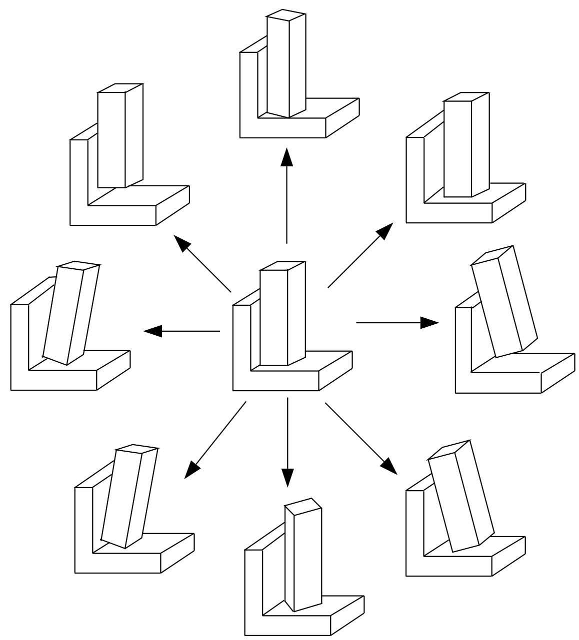


Figure 2: Examples where only infinitesimal transition motions are needed

lower DOF with more complex CF) is quite helpful because it further reduces the complexity of the CF-compliant motion planning problem, in addition to the already reduced DOF and reduced scope of motion because of the CF constraints. In [12, 13], we described novel strategies for random sampling of CF-compliant configurations, which can be used by a randomized planner, such as a planner based on probablistic road maps [17, 18, 19] or evolutionary computation [10, 28] to plan CF-compliant motions of high DOF.

In reality, an infinitesimal transition is more meaningful in *breaking* a contact rather than making a contact, because it is hard to separate the infinitesimal transition from the finite reconfiguration in the latter. Breaking contacts happen during a motion from one contact state to its LCN contact state, which we call a *neighboring relaxation motion*. Note that planning neighboring relaxation motion is a main issue in generating a GCR graph (Section 3.1), and it is also a further simplified problem of the general **Subproblem 2**. This is because in many cases, a neighboring relaxation motion consists of only an infinitesimal transition (Fig. 2), and to seach a suitable infinitesimal transition is an easier problem.

As an infinitesimal motion is characterized by the axis of the motion and the direction of velocity, such a transition motion can often be specified *topologically* in terms of the contact elements involved in the change, either completely or partially [11]. Once the intended motion is specified, its feasibility can be easily checked. In [11], we described how to do the feasibility check based on local CF information, which is applicable to both contacting polygons and contacting polyhedra.

Once suitable CF-compliant reconfiguration motions and infinitesimal transition motions are found, they can be merged to form a suitable motion plan between neighboring contact states, i.e., to solve the *Subproblem 2*. This, in turn, helps solving the **Subproblem 1** (as described in Section 3.1). Moreover, once a contact state graph is created, graph search can be used to yield a suitable sequence of contact states — a high-level plan. Guided by this sequence, neighboring contact motions can be concatenated automatically to form a global contact motion.

3.3 Handling CFs with multiple connected regions

As described in Section 2.2, depending on object geometry, the geometric representation GeoCF of a CF may consist of more than one CF-connected region, and each CF-connected region represents a separate contact state. Fortunately, except for rare cases involving degereate CFs, we can avoid dealing with multiple CF-connected regions during the generation of a single GCR graph and only encounter the issue during merge of graphs: we have pointed out in Section 3.1 that during the merge of GCR graphs, it is necessary to check whether two contact states with the same CF are separate contact states or can be merged as one contact state. We have also begun to study how the geometrical characteristics of contacting objects relate to disconnectedness of a CF in order to quickly determine if a CF has disjoint contact states, or in other words, if two states under the same CF can be merged without going through the search of a CF-compliant motion connecting the states [14].

3.4 Discussion of complexity

We can capture the *topological* complexity of two contacting polyhedra in terms of their numbers of faces, M and N, but these numbers cannot convey the *geometrical* complexity of the objects. Such complexity determines the possible kinds and number of contact states and thus affects contact motion planning more significantly. The arbitrariness of geometric shapes, however, makes it difficult to characterize geometrical complexity in general. This, in turn, makes it difficult to assess the time complexity of our approach in a near-accurate form. In addition, the complexity also depends on the motion planner used for CF-compliant motion planning. Nevertheless, both our reasoning and implementation results (next section) demonstrate the feasibility and advantages of the divide-and-merge approach.

4 Some Implementation Results

We have implemented algorithms for automatic generation and merging of GCR graphs. We limited our current attention to generating GCR graphs for arbitrary CFs with ≤ 3 PCs between arbitrary polyhedra[5]. This is because a CF is fully constrained, i.e., with zero DOF, if its PCs can provide 6 independent constraint equations, which can often be achieved by a 3-PC CF of the right combination, such as one with three f-f PCs, three f-e PCs, or, two f-f PCs and one f-v PCs, and so on, with non-parallel contacting features. Although theoretically, if all PCs are of f-v or e-e-cross types, at most 6 PCs are needed to fully constrain a CF, in many practical cases, a CF with more than 3 PCs often involve redundant PCs and can be found to be equivalent to a CF with fewer PCs [29].

As our emphasis is on the novel aspects of the divide-and-merge approach, our current implementation only employs a simple motion planner which provides feasible straight-line motions for CF-compliant reconfigurations.

We also used a display program written by David Johnston of our group to view the contact state graph and the contact states between polyhedra. It is based on openGL, runs both on UNIX and Windows NT, and allows user interaction to choose what to view and in what fashion.

Fig. 3 shows one example, where Node 1 (the top node) indicates the seed contact state. The GCR graph has 90 nodes, of which some of the contact states are shown (by clicking the nodes).

Fig. 4 shows the result of merging two GCR graphs. The two nodes on the top of the window are the seed nodes of the GCR graphs respectively.

We have run our program on nearly 30 different examples involving polyhedra pairs of vastly different shapes. Figure 5 summarizes information on some of the example GCR graphs generated, run on a SUN Ultra 10 workstation (which is rated at 12.1 SPECint95 and 12.9 SPECfp95).

[5]This extends the results for only polygons reported in [11].

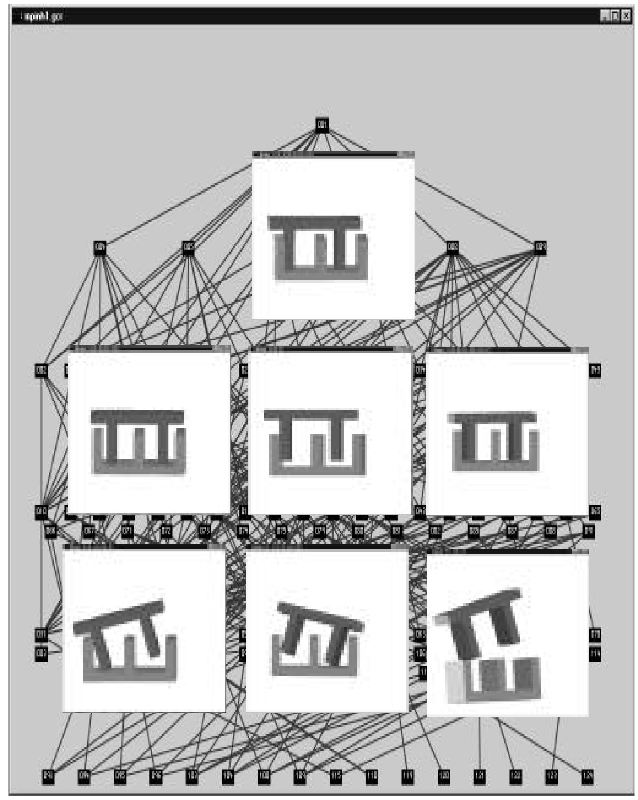


Figure 3: A GCR graph between two polyhedra

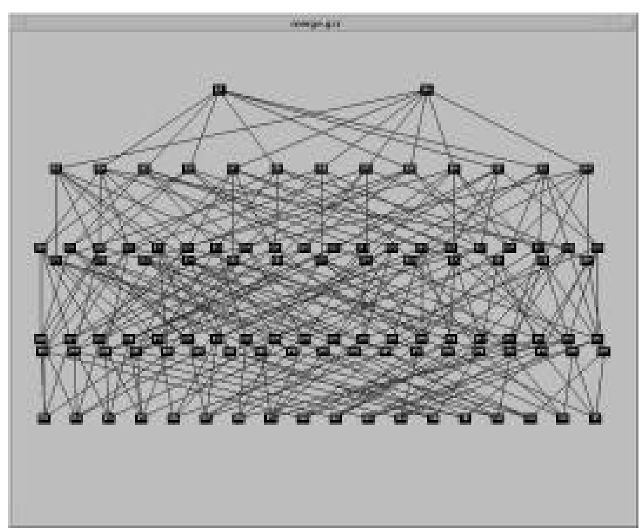


Figure 4: Result of merging two GCR graphs

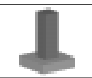
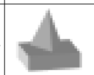
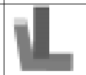
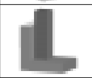
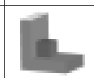
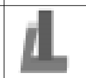
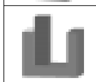
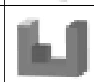
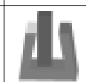
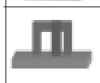
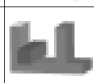
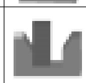
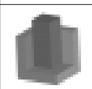
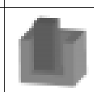
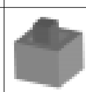
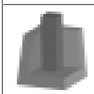
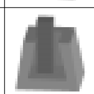
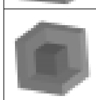
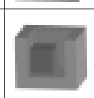
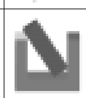
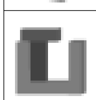
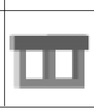
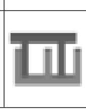
#PC	Seed CF	#nodes in GCR	time (s)	Seed CF	#nodes in GCR	time (s)	Seed CF	#nodes in GCR	time (s)
1PC		81	0.24		63	0.21		54	0.5
		134	3.3		163	2.2		86	1.2
2PC		107	3.8		136	2.3		59	1.2
		176	16.6		124	2.9		114	5.3
		197	16.5		155	13.3		121	11.2
3PC		96	3.5		64	2.7		41	1.8
		249	7.5		197	6.6		67	1.6
		137	9.3		252	5.1		124	19.1

Figure 5: Several experimented examples

5 Conclusion

To conclude, we have introduced a general divide-and-merge approach for planning contact constrained motions. The approach decomposes the hard problem into manageable subproblems. We have implemented the novel aspects of the approach concerning automatic generation of contact state graphs to demonstrate its feasibility. We have also conducted related research on random sampling of contact configurations constrained by a CF [12, 13] and on analysis of geometric complexity [14]. There are a number of further research issues, including effective selection of seed contact states for GCR graphs, more powerful CF-compliant motion planning, and dealing with CFs with many PCs.

Acknowledgement

We would like to thank David Johnston for writing the interactive display program to show contact state graphs.

References

[1] H. Asada and S. Hirai, "Towards a Symbolic-Level Force Feedback: Recognition of Assembly Sequence States," *Int. Symp. Robotics Res.*, pp. 290-295, 198 9.

[2] F. Avnaim, J. D. Boissonnat, and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles," *1988 ICRA*, pp. 1656-1660, April 1988.

[3] Randy C. Brost, "Computing Metric and Topological Properties of Configuration- Space Obstacles," *1989 ICRA*, pp. 170-176, May 1989.

[4] H. Bruyninckx, J. Schutter, "Modeling and Specification of Compliant Motions with Two and Three Contact Points," *IEEE Int. Conf. Robotics & Automation*, pp. 1938-1943, May 1998.

[5] S. J. Buckley, "Planning Compliant Motion Strategies," *Int. J. Robotics Res.*, 8(5), 28-44, 1989.

[6] G. Dakin, and R. Popplestone, "Simplified Fine-Motion Planning in Generalized Contact Space," *IEEE Int. Symp. on Intell. Control*, pp. 281-287, 1992.

[7] R. Desai, *On Fine Motion in Mechanical Assembly in Presence of Uncertainty*, Ph.D. thesis, Department of Mechanical Engineering, the Univ. of Michigan, 1989.

[8] B. R. Donald, "On Motion Planning with Six Degree of Freedoms: Solving the Intersection Problems in Configuration Space," *1985 ICRA*, pp. 536-541, 1985.

[9] H. Hirukawa, "On Motion Planning of Polyhedra in Contact," *1996 WAFR*, 1996.

[10] C. Hocaoglu, and A. Sanderson, "Evolutionary Path Planning Using Multiresolution Path Representation," *1998 ICRA*, pp.318–323, May 1998.

[11] X. Ji and J. Xiao, "Automatic Generation of High-Level Contact State Space," *1999 ICRA*, pp. 238-244, May 1999.

[12] X. Ji and J. Xiao, "Towards Random Sampling with Contact Constraints," accepted to ICRA2000.

[13] X. Ji and J. Xiao, "Random Sampling of Contact Configurations in two-PC Contact Formations,"

accepted to the 2000 Int. Workshop on Algorithmic Foundations of Robotics, Dartmouth College, March 2000.

[14] D. Johnston and J. Xiao, "On Relating the Disconnectedness of a Topological Contact State to the Geometric Properties of its Constituent Objects," accepted to ICRA2000.

[15] L. Joskowicz, R. H. Taylor, "Interference-Free Insertion of a Solid Body Into a Cavity: An Algorithm and a Medical Application," *Int. J. Robotics Res.*, 15(3):211-229, June 1996.

[16] S.C. Kang, et al., "A Compliant Motion Control for Insertion of Complex Shaped Objects using Contact," *1997 ICRA*, pp. 841-846, May 1997.

[17] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Trans. Robotics & Automation*, 12(4):566-580, 1996.

[18] L.E. Kavraki and J.C. Latombe, "Probabilistic Roadmaps for Robot Path Planning," *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, pp. 33-53, 1998.

[19] C. Nissoux, T. Simeon, and J. P. Laumond, "Visibility based probabilistic roadmaps," *1999 IROS*, Kyongju, Korea, Nov. 1999.

[20] T. Lozano-Pérez, "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Comput.*, C-32(2):108-120, 1983.

[21] B. J. McCarragher, "Task Primitives for the Discrete Event Modeling and Control of 6-DOF Assembly Tasks," *IEEE Trans. Robotics and Automation*, 12(2):280-289, April 1996.

[22] J. Rosell, "Fine Motion Planning for Robotic Assembly under Modeling and Sensing Uncertainties," Ph.D. Thesis, Universitat Polytècnica de Catalunya, June 1998.

[23] E. Sacks, C. Bajaj, "Sliced Configuration Spaces for Curved Planar Bodies," *Int. J. Robotics Res.*, 17(6):639-651, June 1998.

[24] S. Shekhar, O. Khatib, "Force Strategies in Real-Time Fine Motion Assembly", *Proc. ASME Winter Annual Meeting*, Boston, 1987.

[25] R. H. Sturges, and S. Laowattana, "Fine Motion Planning through Constraint Network Analysis", *IEEE Int. Conf. Assembly and Task Planning*, pp. 160-170, Pittsburgh, August 1995.

[26] J. Xiao and R. Volz, "On Replanning for Assembly Tasks using Robots in the Presence of Uncertainties," *1989 ICRA*, pp. 638-645, May 1989.

[27] J. Xiao, "Automatic Determination of Topological Contacts in the Presence of Sensing Uncertainties," *1993 ICRA*, pp. 65-70, May 1993.

[28] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive Evolutionary Planner/Navigator for Mobile Robots," *IEEE Trans. on Evolutionary Computation*, 1(1):18-28, April 1997.

[29] J. Xiao and L. Zhang, "Contact Constraint Analysis and Determination of Geometrically Valid Contact Formations from Possible Contact Primitives," *IEEE Trans. Robotics and Automation*, 13(3):456-466, June 1997.

[30] J. Xiao, "Goal-contact Relaxation Graphs for Contact-based Fine Motion Planning," *1997 IEEE Int. Sym. Assembly and Task Planning*, pp. 25-30, Aug. 1997.