An Interactive Approach to Determining Complex Contact States

Wusheng Chou
IMI lab, Dept. of Computer Science
University of North Carolina - Charlotte
Charlotte, NC 28223, USA
wschou@uncc.edu

Jing Xiao
IMI lab, Dept. of Computer Science
University of North Carolina - Charlotte
Charlotte, NC 28223, USA
xiao@uncc.edu

Abstract

Information of topological contact states is needed for a wide range of tasks, especially high-precision robotic assembly tasks, where contact states are inevitable. Often an assembly motion plan involves a sequence of contact state transitions between the part held by a robot and another part to form the assembly or subassembly. Except for the simplest cases, obtaining the information of all possible geometrically valid contact states is far from trivial. There is considerable research on how to obtain all valid contact states and their adjacency graph automatically between polyhedral objects, given certain seed contact states [1]. More recently, how to obtain such information between general curved objects is also considered [2]. However, one open problem is how to obtain the seed contact states in the first place, which are preferred to be locally most constrained contact states involving multiple contact regions. This is especially challenging between general curved objects with boundaries of high-order algebraic surfaces. In this paper, we describe how a fast interactive approach based on accurate real-time distance computation and taking advantage of haptic rendering is particularly suitable for determining complex contact states between general curved objects accurately, which can be used as a handy and accurate tool to find seed contact states for subsequent automatic generation of other possible contact states. We demonstrate the effectiveness of this tool with implemented examples.

1 Introduction

Many robotic tasks including assembly tasks cannot avoid contacts and compliant motion between a robot or the object held by a robot and the environment. Compliant motion planning and control requires the knowledge of contact topology and geometry, as characterized by certain discrete contact states and state transitions [3]. Such knowledge is often manually extracted and fed into a system as input, which can be extremely tedious, incomplete, and error prone for even tasks of simple geometry involving only polyhedral objects [4] and is practically infeasible for complex tasks due to the huge number of complex contact states.

To address the problem, Xiao and Ji [1] developed a systematic approach to generate automatically the contact state space between two arbitrary polyhedral objects in terms of a contact state graph. Each node in the graph denotes a contact state, described by a *contact formation* (CF) [5] and a representative configuration of the CF, and an arc between two nodes indicates feasible neighboring state transitions. Although this approach can generate a large number of contact states and their adjacency information automatically, it requires the input of certain *seed* contact states as the starting points. A seed contact state is preferred to be a locally most constrained contact state so that its neighboring contact states can be generated automatically and efficiently through local contact relaxation motions. Such individual seed contact states are usually constructed manually.

There exist some attempts to generate a contact configuration between two objects which are initially not in an exact contact configuration (i.e., either they are separate or in penetration) automatically by minimizing the distance or penetration distance between the two objects. Goeree *et al.* [6] determine a contact state between two polyhedral parts by searching for a contact configuration that satisfies the contact state using such a numerical optimization approach. Pan *et al* [7] describe an automatic means of obtaining a high-level description of the contact states that may occur during the assembly of two polyhedral objects by extending the work of [5] and [6]. However, these methods based on numerical optimization is time consuming even for polyhedral objects, and its effectiveness is subject to whether the initial relative configuration between the two polyhedral objects is chosen properly so that local optima can be avoided.

More recently, progress is made on how to characterize, represent, and automatically generate topological contact states between general curved objects [2][8]. Since the boundaries of these objects are high-order algebraic curves and surfaces, it is all the more challenging to determine a valid seed contact state or configuration between two such objects, which imply solving complex non-linear equations. Try to determine a valid complex contact state involving multiple contact regions can become simply infeasible.

In this paper, we introduce an approach for fast and interactive determination of valid complex contact states taking advantage of real-time distance computation and haptic rendering. The paper is organized as follows: Section 2 reviews representation of contact states between general curved objects. Section 3 introduces our interactive approach to determining complex contact states. Section 4 provides some implemented examples, and Section 5 points out additional applications of the proposed approach. Section 6 concludes the paper.

2 Contact states between general curved objects

Contact state representations introduced in the literature are mainly for contacting polyhedral objects. It is common to describe a contact state as a set of contact primitives. Each contact primitive defines a single connected region of contact and can be naturally characterized by a pair of contacting surface features in terms of faces, edges, and vertices, because such surface features are convex for polyhedral objects and there can only be one connected region of contact between two such features. Different contact state representations essentially differ only in how contact primitives are determined with those surface features; some define a contact primitive as a point contact in terms of a vertex and a face in contact [9][10], while others allow a contact primitive to be a line or planar contact region as well [5][11].

In order to describe contact states between two non-polyhedral or curved objects in a way analogous to that between polyhedral objects, however, it is necessary to further partition the boundary surface of each non-polyhedral object to obtain surface features of useful properties in addition to the natural partition by the natural edges and vertices that indicate derivative discontinuous points [8]. Specifically, given a general curved object, its boundary consists of smooth surfaces (which includes flat surfaces as a special case), smooth curves (which includes straight-line segments as a special case), and vertices, which are first-derivative discontinuous points separating two smooth curves. Since a smooth surface or curve can be non-convex generally, a contact between two such surfaces or curves may consist of more than one disjoint region of contact. Thus, such smooth surfaces and curves cannot be used directly to define contact primitives.

In [8], it shows that by doing a curvature monotonic segmentation to the boundary surface of a curved object, useful topological surface features can be obtained, which are in terms of smooth surface patches or curve segments with monotonically changing or constant curvatures, separated by curvature extreme points or inflection points of a surface, in addition to vertices. We can label the resulting *topological surface features* of a curvature monotonic segmentation as the following:

- *face*: a monotonic smooth surface patch,
- *edge*: a smooth curve (or straight-line) segment bounding a face that is part of natural edge,
- *pseudo edge:* a smooth curve (or straight-line) segment bounding a face that is not part of a natural edge,
- *vertex:* a natural vertex bounding an edge or a face,
- *pseudo vertex:* an end point of an edge or a pseudo edge that is not a natural vertex.

As an example, Fig. 1 shows the result of curvature monotonic segmentation of an ellipsoid, where there are 8 faces, 12 pseudo edges, and 6 pseudo vertices.

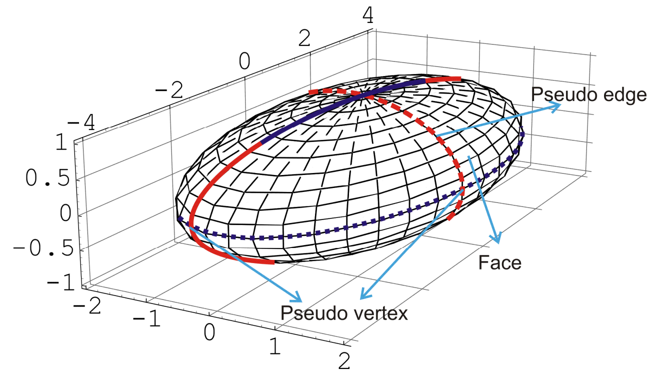


Fig. 1. Curvature monotonic segmentation of an ellipsoid

A contact primitive between two general curved objects can now be defined as a contact between two such surface features, which forms a single connected region of contact, called a *principal contact* (PCs), in analogy to the principal contacts defined for polyhedral objects [8]. A topological contact between two general curved objects can thus be represented in general as a set of PCs formed, called a *contact formation* (CF).

Finally, a *topological contact state* between two general curved objects can be characterized in terms of the corresponding contact formation *CF* and a representative contact configuration *C* that satisfies the contact formation, denoted as *CS* = <*CF*, *C*>.

3 Interactive approach to determining contact states

To determine a valid contact state between two general curved objects accurately is a very challenging problem, especially if the contact state involves multiple PCs. For example, to determine a valid contact state between an ellipsoid and a parabolic bowl (see Fig. 3), we need to solve the non-linear equations of the ellipsoid and the paraboloid for intersection points and to further find the contact points where the tangent planes of the two objects are coincident. Manual derivation of analytical solutions is tedious, may not always be possible for general cases, and is often infeasible for obtaining a large number of different contact states. Numerical optimization methods also may not always find the correct solutions due to local optima, are subject to the selection of initial starting points, and are often time consuming.

We propose to solve the problem by an interactive approach via haptics, taking advantage of the recent progress in real-time exact computation of minimum distances between two general curved objects.

Specifically, given the geometric models of two curved objects A and B, which also include information of topological surface features, our approach is to allow a human user to create different contact situations between the two models of the objects displayed in a virtual environment, where the user virtually holds one object A via a haptic device to contact the other object B. As the user manipulates A and creates contacts between A and B, a real-time distance computation algorithm continuously runs to provide instant information of how close A is to B and where the minimum distances occur.

A contact state between A and B happens when the minimum distances between A and B are all within certain very small threshold (taking into account inevitable digital errors). When a contact is detected, our algorithm provides the pairs of features in contact, i.e., the principal contacts (PCs), the exact contact points, as well as the exact contact configuration of A relative to B. Thus, an exact contact state is obtained.

To enable easy creation of contact states between A and B while avoiding interpenetrations between the two virtual object models, simple contact normal forces are simulated via the haptic device to provide the user force feedback. When the minimum distance d_i between two features of A and B is less than a threshold ε (> 0), we consider the two features form a potential principal contact PC_i. Let F_i denote the magnitude of the contact normal force at PC_i. We simply model F_i based on a spring model as:

$$F_i = K(\varepsilon - d_i).$$

F_i is not too large with a relatively small K to ensure rendering stability, but it is sufficient to prevent penetration of A and B as d_i becomes even smaller.

When d_i becomes smaller than an even smaller threshold $0 < \delta < \varepsilon$, our program could change the colour of the related features of A and B to indicate that a contact indeed occurs, i.e., PC_i happens.

We use the novel hybrid method recently introduced by Chou and Xiao [12] for real-time exact distance computation between two general, non-convex curved objects where multiple contacts can occur at the same time. This method is characterized by the following unique advantages:

- It builds a hybrid representation of objects of different models in pre-processing to best facilitate fast on-line distance/contact computation that combines the powers and merits of these models.

  Specifically, the method establishes bounding volume trees for components of general objects and dual representations for each component in terms of both polygonal meshes and finite surface features described by exact parametric equations.

- Its on-line procedure is a hybrid algorithm taking advantage of three different stages/methods to maximize both the efficiency and accuracy of on-line distance/contact computation between two general objects that are close by.

  The hybrid algorithm combines relatively coarse intersection checking based on bounding-volume hierarchies when objects are far apart and minimum distance computation when they are sufficiently close. For the latter, it first performs fast approximate distance computation based on polygonal mesh models and then switches to exact distance computation based on accurate parametric representation of surface features. As the result, the algorithm can provide in real-time the pairs of surface features in contact and the exact pairs of contact points.

- Its performance both in terms of efficiency and accuracy is relatively independent of both the resolution of polygonal mesh models and the number of surface features of the objects in parametric representations, and therefore the algorithm is robust and highly scalable.

Experiments show that this method has an update rate from several hundred Hz for detecting very complex contacts to *k*Hz, which is among the fastest in contact detection algorithms.

With our interactive approach described above, one can find valid seed contact states quickly, easily, and accurately, which can be used for automatic generation of contact state graphs [1][2]. Moreover, this interactive approach enables a user to incorporate naturally human intelligence in identifying the seed contact states and the corresponding neighborhoods (or subgraph) of contact states most relevant to the execution of a particular assembly task and in detecting the most likely contact states and contact state transitions. Such knowledge is invaluable for the generation of effective assembly motion plans.

4 Implemented examples

We have implemented our approach in Visual C++ and run it on a Pentium Processor of 2.8GHz CPU with 1GB RAM. A human operator can hold a virtual object A to make arbitrary contact to another virtual object B freely via a PHANToM Desktop device. The following provides the values of the parameters used in our experiments, where m is the mass of A.

Table I: Parameters and their values

Parameter	Value
m	0.5 (Kg)
K	2.3 (N/mm)
ε, δ	1.0 (mm), 0.01 (mm)

We define a frame for each object. The frame of the fixed object B is coincidence with the world frame, and the origin of the frame of the held object A locates at its geometric center. Let O denotes the frame of A, and W denotes the frame of B.

In our first example, A is an ellipsoid and B is a bowl with a parabolic inner surface. Let (x_A, y_A, z_A) denote a point with respect to A's frame and (x, y, z) denote the coordinate of a point with respect to the B's frame (which is the world frame). The algebraic descriptions of the ellipsoid and the parabolic surface in their respective frames are shown as equations (1) and (2), respectively.

$$4x_A^{\ 2} + 16y_A^{\ 2} + 4z_A^{\ 2} = 1 \quad (1)$$
$$4x^2 + 4y^2 = z \quad (2)$$

When A is undergoing motion, the configuration of A at any instant can be described by the homogeneous transformation matrix ${}^{W}T_O$ from A to B.

$${}^{W}T_O = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_0 \\ r_{21} & r_{22} & r_{23} & y_0 \\ r_{31} & r_{32} & r_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The inverse matrix of ${}^{W}T_O$ is:

$${}^{O}T_W = {}^{W}T_O^{\ -1} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & -(r_{11}x_0 + r_{21}y_0 + r_{31}z_0) \\ r_{12} & r_{22} & r_{32} & -(r_{12}x_0 + r_{22}y_0 + r_{32}z_0) \\ r_{13} & r_{23} & r_{33} & -(r_{13}x_0 + r_{23}y_0 + r_{33}z_0) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Thus the equation of the ellipsoid in terms of coordinates with respect to the world frame is shown as equation (5).

$$\begin{aligned} &4[r_{11}(x-x_0) + r_{21}(y-y_0) + r_{31}(z-z_0)]^2 \\ &+16[r_{12}(x-x_0) + r_{22}(y-y_0) + r_{32}(z-z_0)]^2 \\ &+4[r_{13}(x-x_0) + r_{23}(y-y_0) + r_{33}(z-z_0)]^2 = 1 \end{aligned} \quad (5)$$

Note that the parameters in equation (5) are changing during the motion of the ellipsoid. Obviously, to obtain intersection points between A and B by solving equations (2) and (5) is quite complex, and to obtain contact points between A and B further complicates the computation.

Using the interactive approach, we can conveniently obtain the contact feature pairs as well as exact contact points between A and B. A has 26 features: 6 pseudo vertices labeled PV_{A1}–PV_{A6}, 12 pseudo edges labeled PE_{A1}–PE_{A12} and 8 elliptic faces labeled F_{A1}–F_{A8} (see Fig.1). The inside of the parabolic bowl has 9 features: 3 pseudo vertices labeled PV_{B1}–PV_{B3}, 2 edges E_{B1} and E_{B2}, 2 pseudo edges PE_{B1} and PE_{B2}, and 2 faces F_{B1} and F_{B2} (see Fig. 2).

Fig. 3 shows some contact states between the ellipsoid and the inside of the parabolic bowl obtained via the interactive approach, and each contact state can consist of one or two principal contacts (PCs). Fig. 3(e) shows a 2-PCs contact state that involves a pseudo vertex of the ellipsoid along its major axis; this can be viewed as a seed contact state for it has the least degree of freedom of motion (sliding or rolling).

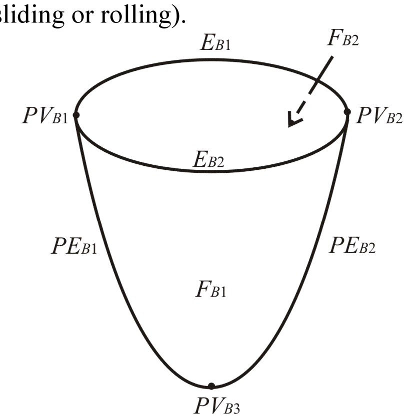
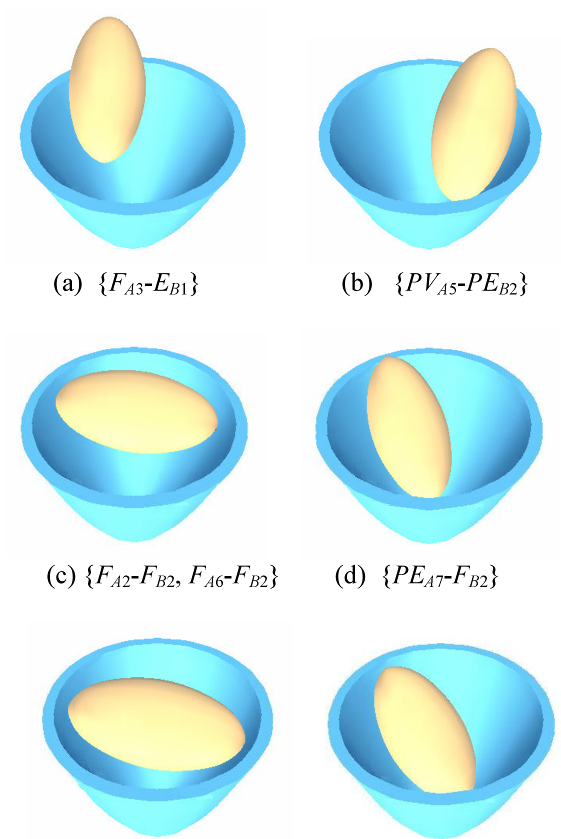


Fig. 2. The features of a parabolic bowl



(a) $\{F_{A3}\text{-}E_{B1}\}$ (b) $\{PV_{A5}\text{-}PE_{B2}\}$

(c) $\{F_{A2}\text{-}F_{B2}, F_{A6}\text{-}F_{B2}\}$ (d) $\{PE_{A7}\text{-}F_{B2}\}$

(e) $\{F_{A7}\text{-}F_{B2}, PV_{A1}\text{-}F_{B1}\}$ (f) $\{PE_{A3}\text{-}F_{B2}, F_{A6}\text{-}F_{B1}\}$

Fig. 3. Some contact states obtained via the interactive approach between an ellipsoid and a parabolic bowl

Our interactive approach avoids the tedious effort to derive the exact contact points. For instance, we can easily get two pairs of contact points for the contact state shown in Fig. 3(e):

- p_A (-0.5022, 0.0416, 1.071) and
  p_B (-0.5155, 0.0416, 1.070)

- q_A (0.4843, -0.0935,1.016) and
  q_B (0.4948, -0.0934,1.014)

Note that q_A is corresponding to a pseudo vertex of the ellipsoid, and its coordinate is (0.5, 0.0, 0.0) with respect to the local frame of the ellipsoid. In Fig. 3(e), the transformation matrix of the ellipsoid with respect to the world frame is shown as equation (6):

$$ {}^{W}T_{O} = \begin{bmatrix} 0.9856 & 0.1208 & 0.1178 & -0.0101 \\ -0.1434 & 0.9678 & 0.2068 & -0.0219 \\ -0.0891 & -0.2207 & 0.9713 & 1.0601 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

By substituting p_A , p_B ,q_A and q_B into equation (5) and equation (2), we can see that these two pairs of contact points satisfy both equation (5) and equation (2), with the computation error in the order of 10^{-4}.

Another implemented example is the multiple peg-in-hole insertion task, where object *A* consists of three pegs, two round and one elliptic, and object *B* consists of three holes, two round and one elliptic. The world frame (i.e., *B*'s frame) origin locates at the center of the top planar surface of *B*. The origin of *A*'s frame locates at the center of the three pegs. The holes are represented in the world frame as shown in equations (7)–(9), and the pegs are represented in *A*'s frame as equations (10)–(12) respectively.

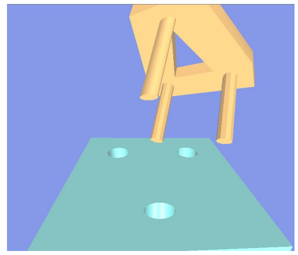
$$(x-5)^2+(y-5)^2 = 9, \quad (0 \le z \le 5) \qquad (7)$$
$$(x+5)^2+(y-5)^2 = 9, \quad (0 \le z \le 5) \qquad (8)$$
$$x^2/1.5^2+(y+6)^2 = 1, \quad (0 \le z \le 5) \qquad (9)$$
$$(x_p-5)^2+(y_p-5)^2 = 4, \quad (0 \le z_p \le 10) \qquad (10)$$
$$(x_p+5)^2+(y_p-5)^2 = 4, \quad (0 \le z_p \le 10) \qquad (11)$$
$$x_p^2+(y_p+6)^2/0.5^2 = 1, \quad (0 \le z_p \le 10) \qquad (12)$$

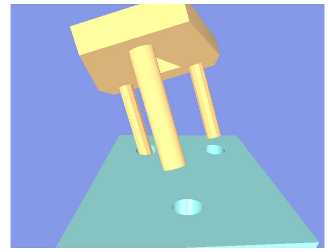
We again can obtain different contact states with exact contact feature pairs and corresponding contact points between *A* and *B* through our interactive approach efficiently and effectively. Fig. 4 shows some achieved contact states during the insertion task.

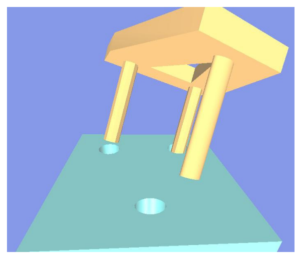
The average time for the detection of a contact state in our experiments is in the order of a few milliseconds, which is appropriate for real-time haptic rendering.

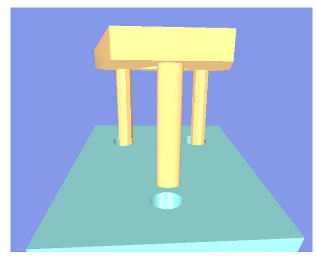
5 Evaluation of assembly plans via virtual prototyping

Our interactive approach not only provides an efficient and accurate method to determine complex contact states between general objects to facilitate assembly or compliant motion planning but also enables evaluation of assembly plans via virtual prototyping. Recently Luo and Xiao [13] introduced an efficient method for high-fidelity haptic rendering of contact force and moment given a contact state in real time, taking into account friction, gravity and dynamic effects. The method allows realistic rendering of compliant motion effect via haptics. By coupling such a method with our real-time method of determining a contact state accurately, realistic simulation of assembly motion in a virtual task environment, i.e., virtual prototyping, can be achieved. The haptics-based simulation environment allows a human operator to accomplish virtual assembly tasks with high-fidelity force/moment feedback to evaluate the effectiveness of different assembly motion plans.


(a) no peg in hole


(b) one peg in a hole


(c) one peg in a hole


(d) two pegs in holes


(e) three pegs in holes

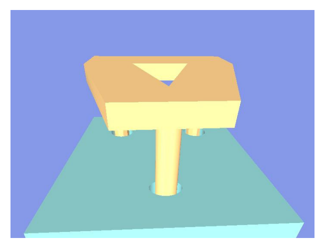

(f) three pegs in holes

Fig. 4. Evaluation of assembly plans of the multiple peg-in-hole insertion task

6 Conclusions

We have introduced an interactive approach to determining complex contact states between general, non-convex curved objects that takes advantage of fast and accurate real-time distance computation and haptic rendering. Using this approach, the user can naturally and intuitively manipulate virtual object via a haptic device and find valid seed contact states efficiently and accurately, which can be used for automatic generation of contact state graphs. Furthermore, this interactive approach enables a user to incorporate naturally human intelligence in identifying the most relevant seed contact states and the corresponding neighborhoods of contact states to the execution of a particular assembly task and in detecting the most likely contact states and contact state transitions. Creation and evaluation of assembly plans can also be done by incorporating the proposed approach. One of our on-going projects is to create a high-fidelity virtual assembly simulation and evaluation environment.

Acknowledgement

The authors are grateful to the support of National Science Foundation of U.S. under grants IIS-0328782 and EIA-0203146.

References

[1] J. Xiao and X. Ji, "Automatic generation of high-level contact space," *International Journal of Robotics Research*, Vol. 20, No. 7, pp. 584-606, 2001.

[2] P. Tang and J. Xiao, "Automatic Generation of High-level Contact State Space between Planar Curved Objects," submitted to *IEEE Int. Conf. on Robotics and Automation,* 2005.

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

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

[5] J. Xiao, "Automatic Determination of Topological Contacts in the Presence of Sensing Uncertainties," *IEEE Int. Conf. Robotics & Automation*, pp. 65-70, May 1993.

[6] B. B. Goeree, E. D. Fasse, and M. M. Marefat, "Determining feasible contact state of pairs of spatial polyhedra," *IEEE Int. Conf. on Robotics and Automation*, pp. 1396-1401, 2000.

[7] F. Pan and J.M. Schimmels, "Efficient Contact State Graph Generation for Assembly Applications," *IEEE Int. Conf. on Robotics and Automation*, pp. 2582-2598, 2003.

[8] Q.Luo, E. Staffetti, and J. Xiao, "On the representation of contact states between curved objects," *IEEE Int. Conf. on Robotics and Automation*, pp. 3589- 3595, 2004.

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

[10] B. R. Donald, "On Motion Planning with Six Degree of Freedoms: Solving the Intersection Problems in Configuration Space," *IEEE Int. Conf. Robotics & Automation*, pp. 536-541, 1985.

[11] R. Desai, "On Fine Motion in Mechanical Assembly in Presence of Uncertainty," Ph.D. thesis, Dept. of ME, Univ. of Michigan, 1989.

[12] W. S. Chou and J. Xiao, "A Hybrid Approach for Accurate, Real-time Contact Detection from Distance Computation between General Curved Objects," submitted to *IEEE Trans. on Robotics*.

[13] Q. Luo and J. Xiao, "Physically Accurate Haptic Rendering with Dynamic Effects," *IEEE Computer Graphics and Applications*, pp.60-69, 2004.